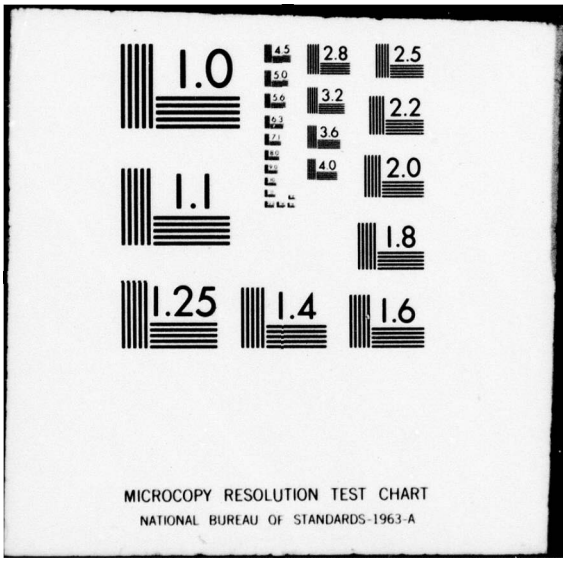
AD-A065 888 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
SEP 78

UNCLASSIFIED CCTC-TR-54-78 SBIE-E100 124 NL

1 OF 2
AD
A065 8


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TECHNICAL REPORT
TR 54-78
1 SEPTEMBER 1978

(12)

LEVEL III

C
C
T
C

COMMAND
& CONTROL
TECHNICAL
CENTER

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

INSTALLATION OF NIPS 360 FFS

DEFENSE
COMMUNICATIONS
AGENCY

(18) SBIE (19) E100 124

COMMAND AND CONTROL TECHNICAL CENTER

(9) Technical Report Number TR 54-78

(11) 1 September 1978

(6)

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

Installation of NIPS 360 FFS

(12) 174 p.

(14) CCTC-TR-54-78

SUBMITTED BY:

CRAIG K. HILL
Captain, USA
CCTC Project Officer

APPROVED BY:

FREDERIC A. CRAF[illegible], JR.
Captain, U.S. Navy
Deputy Director
NMCS ADP

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314

This document has been approved for public release and sale; its distribution is unlimited.

AD58 865

409 658 79 03 08 018 slt

ACKNOWLEDGMENT

This manual was prepared under the direction of the Chief for Programming with general support provided by the International Business Machines Corporation under contracts DCA 100-67-C-0062, DCA 100-69-C-0029, DCA 100-70-C-0031, DCA 100-70-C-0080, DCA 100-71-C-0047 and DCA 100-77-C-0065.

CONTENTS

ILLUSTRATIONS

ABSTRACT

This Technical Report describes the installation and maintenance of the NIPS 360 Formatted File System. It describes the hardware and software requirements of the S/360 Operating System (OS). The functions to be performed to install NIPS 360 FFS are presented and a detailed typical 2314 installation is included as a guideline. The procedures for reporting system deficiencies and for periodically updating the system to incorporate system improvements are included.

This document supersedes TR 54-77.

TR 54-78 is part of the following NIPS 360 FFS documentation.

CSM UM 15-78	Vol I	- Introduction to File Concepts
	Vol II	- File Structuring (FS)
	Vol III	- File Maintenance (FM)
	Vol IV	- Retrieval and Sort Processor (RASP
	Vol V	- Output Processor (OP)
	Vol VI	- Terminal Processing (TP)
	Vol VII	- Utility Support (UT)
	Vol VIII	- Job Preparation Manual
	Vol IX	- Error Codes
CSM GD 15-78		- General Description

INSTALLATION

Section 1

INTRODUCTION

This document describes the installation and maintenance of the NIPS 360 Formatted File System.

Section 2 describes the hardware and software requirements of the system.

Section 3 describes the materials included in an Installation Package.

Section 4 describes the functions to be performed to install NIPS 360 FFS. (A detailed typical 2314 Installation is included in appendix B.)

Sections 5 and 6 describe the procedures for reporting system deficiencies and for periodically updating the system to incorporate system improvements.

Section 7 deals with the special requirements of the Terminal Processing (TP) component.

It is assumed that the reader of this document is generally familiar with the IBM System/360 and its Operating System. In addition, a conceptual knowledge of NIPS 360 FFS is desirable. No attempt has been made to explain or define all of the special terminology involved in these systems.

Section 2

SYSTEM DESCRIPTION

The NIPS 360 FFS has been designed to operate on IBM System/360 under the IBM Operating System. This design takes advantage of the upward compatibility of the various computers in the System/360 family and allows device independence among peripheral equipment.

2.1 Hardware Requirements

NIPS 360 FFS has been designed and programmed for an IBM System/360 Model 50H (256K core size). It will also operate on a Model 40H and larger models of the System/360. Without on-line terminals, it will operate on a Model 40G or 50G (128K core size). The NIPS 360 FFS can use magnetic tapes, direct access devices, card reader/punches, on-line printers, a console typewriter, and the following terminals for teleprocessing capabilities:

a. IBM 2260 Display Stations (Local and Remote)

b. IBM 2250 Display Units (Local)

c. IBM 1050 Data Communications Systems (Dial-Up Remote)

d. IBM 2741 Communications Terminals (Nonswitched Mode)

e. IBM 3270 Information Display System (local and remote).

A minimum configuration could include three IBM 2311 Disk Units, a card reader, and an on-line printer, although an IBM 2314 Disk Storage Unit is considered highly desirable. Tape requirements are related to the user's requirements and range from none to a quantity sufficient to perform the largest sort the user may require. Disk sorting

is used to the capacity of the direct access devices available to the system.

The system will service an unlimited number of IBM 2848 Display Control Units; each unit will support one 1053 printer and eight local 2260 terminals for on-line processing. An unlimited number of 2250 terminals are supported under the following configurations:

a. 2250-1 Each terminal has its own control unit

b. 2250-2 Up to four terminals for each 2840-1 control unit

c. 2250-3 Up to four terminals for each 2840-2 control unit

An unlimited number of 3270 terminals/printers are also supported. Each 3270 control unit may have up to 32 terminals/printers attached. Independent 3275 display systems are also supported.

2.2 Operating System Requirements

NIPS 360 FFS will run under three main configurations of OS/360's:

a. Primary Control Program (PCP) - The standard Operating System which processes one job at a time in sequence.

b. MFT-II (Option 2) - Multiprogramming with a fixed number of tasks - a partitioned core system allowing up to four jobs to operate concurrently in independent fixed-size partitions.

c. MVT (Option 4) - Multiprogramming with a variable number of tasks - a regional core system allowing up to 15 jobs to operate concurrently in variable-size regions.

The NIPS 360 FFS is not restricted to any one level of the Operating System and it will use subsequent releases as they become available.

INSTALLATION

NIPS 360 FFS will also run on a System/370 VS1, VS2 Release 1 and VS2 Release 2 version of the operating system.

Multiprogramming capabilities of the Operating System permit multiple jobs to operate concurrently in independent partitions or regions. The OS/360 data management facilities will handle all system and user data storage. NIPS 360 FFS data files can be organized under Indexed Sequential Access Method (ISAM) for disk files or Sequential Access Method (SAM) for tape or disk files. The various system libraries will be stored using the Basic Partitioned Access Method (BPAM). The basic graphic programming services of the Operating System are used for the Terminal Processing (TP) component that services local 2260 and 2250 display terminals. The Basic Telecommunication Access Method (BTAM) is used for remote support of 2260, 1050 and 2741 terminals and support of 3270 terminals.

Users who do not require TP support may operate the NIPS 360 FFS under the Primary Control Program (PCP) option of the Operating System.

In addition to the basic Operating System, the following must be included at system generation (SYSGEN) time:

a. Programs

- ASSEMBLER - F level
- LINKAGE EDITOR
- SORT

b. Libraries

- Macro Library - MACLIB
- Sort Library - SORTLIB

c. Data Management Access Methods

- Sequential Access Method (SAM)
- Direct Access Method (DAM)
- Basic Partitioned Access Method (BPAM)
- Basic Graphics Programming Services (needed only for the TP support)
- Basic Telecommunication Access Method (BTAM) (needed only for specific device support)

o Variable Indexed Sequential Access Method (VISAM)

The following may require that special provisions in the Operating System be made at SYSGEN time depending on installation requirements:

a. NIPS 360 FFS Supervisor Call (SVC) Routine - Installations that operate in S/360, and S/370 VS1 and VS2 Release 1 environments and require NIPS/TP support must reserve a Type III SVC at OS SYSGEN time. The TP MONITOR in the NIPS 360 FFS TP component uses the SVC routine and is distributed assuming the SVC number is 240. The SVC number is not restricted to 240 but will require the using installation to generate a new TP Monitor to recognize an SVC number other than 240. Section 7 of this report contains detailed information on the TP component and its installation. Installations operating S/370 MVS verion of the Operating System do not need the TP SVC for TP support.

b. Generic Unit Names - Generic unit names are used by all NIPS 360 FFS procedures to minimize the number of modifications required to adapt the procedures to an installation's hardware configuration. These generic names are not required but will eliminate the requirement to insert unit numbers in all NIPS 360 FFS procedures. Installations should include the following generic unit names at SYSGEN time: NIPW for disk work space; TAPE9 for 9-track tape; and TAPE7 for 7-track tape.

2.3 S/370 VSAM Requirements

On an IBM System/370, NIPS will perform as on the System/360. However, the user has the option of processing direct access data files using the Virtual Storage Access Method (VSAM). In order to perform VSAM processing, the System/370 must be a Model 135 or larger and have OS/VS (operating system/virtual storage) installed. The system must also have the dynamic address translator. Direct access storage devices which may be used include the IBM 2314, the 2319, the 3330, and the 3333 devices.

Section 3

INSTALLATION PACKAGE

In this section, the NIPS 360 FFS user is given as much material as possible to assist him in installing the system. This not only includes the basic NIPS 360 FFS system, but a set of sample JCL used to install the system and a comprehensive set of test programs to validate a system installation. The NIPS 360 FFS system distributed on magnetic tape will be in one of the two following formats:

a. A dump/restore which will always be on 800 BPI, 9-track, SL tape labeled FFS360

b. Unloaded partitioned and sequential data sets which will always be on 800 BPI, 9-track, SL tape labeled FFS360.

Installations that have 2314's with the track overflow feature will receive a dump/restore tape; all other installations will receive the system as unloaded partitioned and sequential data sets. The dump/restore tape format will be the normal mode of distribution of the system.

3.1 Overview

The NIPS 360 FFS Installation Package consists of:

a. Sample JCL to assist in creating the new system (appendix C)

b. The NIPS 360 FFS system data sets in the form of either a dump/restore tape or unloaded partitioned and sequential data sets

c. A library of sample jobs that exercise each component to ensure that the system installation was successful

d. A library of sample subroutines and tables used by the sample jobs during system testing.

The library of sample jobs and the library of sample subroutines and tables will be distributed as system data sets and will be included on the system tape.

3.2 System Data Sets

NIPS 360 FFS is a disk resident system existing as two data sets: a Program Library (FFS.JOBLIB) containing executable load modules and a Macro Library (FFS.JOBMACRO) containing macros used when NIPS generated code is being compiled. A Skeleton File Library must exist but can occupy a minimum amount of space (i.e., one cylinder). A PTF program library (PTF.JOBLIB) and macro library (PTF.JOBMACRO) must also exist, but will not contain executable load modules or macros, until program transmittal fixes have been applied to the system. Other data sets are included with their purposes defined.

3.3 Program Library - FFS.JOBLIB

The NIPS 360 FFS executable load modules (programs) are in a partitioned data set called FFS.JOBLIB. The members of this library are a result of link editing the NIPS routines and subroutines into appropriate overlay structures. The NIPS system library is referenced on the STEPLIB DD statement and by the JOBLIB symbolic parameter in the NIPS procedures. The Program Library is concatenated to the user's File library and General Library (DUMMY.FILEL) so that subroutines, tables, RITs, and retrievals can be stored on this library although they are usually placed on a User Library. Depending on the type of installation, this library will be received as either an unloaded PDS or as part of a dump/restore tape of the NIPS system disk pack.

3.4 Macro Library - FFS.JOBMACRO

The NIPS 360 FFS generative code macros are in a partitioned data set called FFS.JOBMACRO. These macros are called during the compilation of the generated code produced

by the NIPS components and this library must be on-line whenever NIPS batch jobs are being executed. Depending upon the type of installation this library will be received as either an unloaded PDS or as part of the dump/restore tape of the NIPS system disk pack.

3.5 Procedures Library - FFS.PROCLIB

Each NIPS 360 FFS component is executed using cataloged procedures. These procedures use the symbolic parameter feature which allows the user to run most NIPS jobs with no DD statement overrides. Additional information regarding the use of the NIPS procedures can be found in the "NIPS 360 FFS Job Preparation" manual.

The NIPS 360 FFS procedures are set up with the UNIT parameters specifying NIPW for direct access work space, TAPE9 for 9-track tape, and TAPE7 for 7-track tape. If these parameters are not recognized at an installation the procedures will have to be modified. For example, 2314 may have to be substituted for NIPW.

Depending on the type of installation, this library will be received as either an unloaded PDS or as part of the dump/restore tape of the NIPS system disk pack. The NIPS procedures should be moved into the SYS1.PROCLIB data set. Any procedure modifications should be made before the procedures are moved to SYS1.PROCLIB.

3.6 Skeleton File Library - DUMMY.FILEL

NIPS 360 FFS requires that a Skeleton File Library called DUMMY.FILEL exist although it can occupy a minimum amount of space (i.e. one cylinder). The Skeleton File Library is concatenated to the User Library so that subroutines, tables, RITs and retrievals could be maintained on this library although they are usually placed on a User Library. In this arrangement the Skeleton File Library could be utilized as a generalized library. If DUMMY.FILEL is to be used as a generalized library, the user should be aware that NIPS 360 FFS components <u>will not store</u> subroutines, tables, RITs and retrievals directly into this library. Depending upon the type of installation, this

library will be received as either an unloaded PDS or as part of a dump/restore tape of the NIPS system disk pack.

3.7 Sample Jobs Library - NIPS.SAMPLE.JOB

A library of sample jobs to test the major components of the NIPS system is a partitioned data set consisting of 80 character card images. The sample jobs library is distributed on the system tape.

The sample job is stored on NIPS.SAMPLE.JOB library and consists of 15 members. Each member contains card images that are the NIPS source statements for testing each major component. The test is made up of eight jobs with as many as three steps in one job. The first member of NIPS.SAMPLE.JOB is (BLDJCL). This member contains the job control language which is necessary to execute each job. The SYSIN DD statement for each job contains the member name of the required stored source statements. These members are: DTGOS, SUBLDR, RCMDS, BLDTEST, GENFM, UPDFM, TPTQPSD TSTOPSD, TSTRASP, TSTOP, TSTRAQU, TSTQUIP, TPLOGFFT, TPLOGLS.

The following listing is a copy of the Job Control Language used in this job:

```
//SUBTEST JOB ,FFS,MSGLEVEL=1,PRTY=12
//OUTDATE EXEC ASMFCL
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(DTGOS)
//LKED.SYSLMOD DD DSNAME=&TEMP(DTGOS),DISP=(NEW,PASS),UNIT=SYSDA,
//         SPACE=(CYL,(5,5,200))
//SUBLDR EXEC XSUBLDR,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(SUBLDR)
//TABGEN JOB ,FFS,MSGLEVEL=1,PRTY=11
//TAB2 EXEC XTABGEN,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(RCMDS)
//FFTJOB JOB ,FFS,MSGLEVEL=1,PRTY=10
//FS EXEC XFS,ISAM=TESTER,LIB=TESTER,NDISP=KEEP
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(BLDTEST)
//FM EXEC XFM,ISAM=TESTER,LIB=TESTER,GEN=OVFLOW=3
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(GENFM)
//UPDT EXEC XFMEX,ISAM=TESTER,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(UPDFM)
//QUIPSD JOB ,FFS,MSGLEVEL=1,PRTY=7
```

```
//QUIPSD EXEC XQUIPSD,ISAM=TESTER,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTQPSD)
//OPS  JOB  ,FFS,MSGLEVEL=1,PRTY=6
//OPSD  EXEC XOPSD,ISAM=TESTER,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTOPSD)
//RASPOP  JOB  ,FFS,MSGLEVEL=1,PRTY=5
//GO  EXEC XRASP,ISAM=TESTER,LIB=TESTER,LIBDISP=OLD
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTRASP)
//GOGO EXEC XOP,ISAM=TESTER,LIB=TESTER,LIBDISP=OLD
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTOP)
//RASPQUIP  JOB  ,FFS,MSGLEVEL=1,PRTY=4
//GO  EXEC XRASP,ISAM=TESTER,LIB=TESTER
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTRAQU)
//GOALSC  EXEC XQUIP,SOURCL=TESTER,SDISP=OLD,VSOURCL='SER=FFSLIB'
//QUIP.SOURCLIB DD DSN=TESTERL(QUERY1)
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TSTQUIP)
//TPLOGJOB JOB  ,FFS,MSGLEVEL=1,PRTY=9
//FS EXEC XFS,ISAM=LOGFILE,NDISP=KEEP
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TPLOGFFT)
//FM EXEC XFM,ISAM=LOGFILE
//SYSIN DD DISP=(SHR,KEEP),UNIT=2314,DSN=NIPS.SAMPLE.JOB(TPLOGLS)
```

After the NIPS.SAMPLE.JOB library has been restored to disk, the IEBGENER system utility may be used to punch the member BLDJCL. At this time any required alterations to the JCL may be made by the user to satisfy installation and/or system requirements. All SYSIN DD statements in BLDJCL refer to a member on NIPS.SAMPLE.JOB. The user must alter the JCL to identify the volume serial number NIPS.SAMPLE.JOB or catalog the library before the test is conducted.

To perform the test the user must submit a batch job consisting of the BLDJCL member.

The first job, SUBTEST, will compile and load a user subroutine (DTGOS) onto a file library called TESTERL. This library is supplied with the installation package and contains additional subroutines and tables needed by jobs which test other components (OP, RASP).

The next job called FFTJOB is a three step job that structures the FFT using File Structuring component; generates the file using File Maintenance; and updates the file using FM. In addition to adding data to the file, the FM step compiles logic statements and includes them with the

FFT for the TESTER file. (NOTE: The INDEX, PRIME, and OVFLOW parameters on the execute card of the FM step reflect space considerations for a 2314 file. If the TESTER file is on a 2311, the PRIME and OVFLOW values need only be tripled since the original space estimates were conservative. Installations with 3330 disks may decrease the space allocation accordingly.

Parts of the TESTER file are then displayed by QUIP in the QUIPSD JOB and by the Output Processor in the OPSD JOB.

The RASPOP JOB is a combination of the Retrieval and Sort Processor component and the Output Processor component.

The RASPQUIP job is a combination of RASP and QUIP in a two step job.

The final job (TPLOGJOB) is an example of creating an FFT and a logic statement which could be used to generate and maintain a file containing statistical data on terminal usage.

The source statements for this test were organized by job as individual members of a library for the convenience of the user. If the user prefers, all of the source statements may be output in punched card form and entered into the job stream via the card reader. The number of source statements required is in excess of 3400 punched cards.

3.8 Sample Subroutine Library - TESTERL

A partitioned data set named TESTERL contains subroutines and tables used by the sample jobs. This data set must exist and be cataloged to run sample jobs but it can be scratched once the NIPS 360 FFS has been satisfactorily tested. This data set will be sent as either an unloaded PDS or as part of a dump/ restore tape of the NIPS system disk pack.

3.9 Blocksize Considerations

The system data sets are distributed with the space and blocksize parameters specified in appendix D. NIPS 360 FFS procedures concatenate the SYS1.MACLIB to FFS.JOBMACRO. In order to satisfy concatenation requirements of the Operating System, the blocksize of FFS.JOBMACRO must be greater than or equal to the blocksize of SYS1.MACLIB. Therefore, depending upon the blocksize of SYS1.MACLIB, the blocksize of the FFS.JOBMACRO data set may have to be increased. Likewise, the NIPS 360 FFS procedures concatenate User Subroutine Libraries with the SLIB DD statement such that FFS.JOBLIB is concatenated to a data set having the DCB attributes of SYS1.LINKLIB. This requires that FFS.JOBLIB have a blocksize equal to or less than the block size of SYS1.LINKLIB.

Section 4

INSTALLATION PROCEDURES

To initialize NIPS 360 FFS, four principal functions must be performed using S/360 utilities. These functions are to:

a. Catalog the data sets required by the system

b. Restore the system data sets to a direct access device

c. Move the NIPS 360 FFS procedures (FFS.PROCLIB) to the System Procedures Library - SYS1.PROCLIB

d. Run the sample jobs to ensure a successful system installation.

Each function is discussed in the following paragraphs. An illustrative procedure for installing the NIPS 360 FFS at a computer facility with IBM 2314 disk units is provided in appendix B. This typical procedure shows the type of JCL required by the S/360 utilities to perform the functions described below.

The first function in initializing the system is to catalog the following system data sets: FFS.JOBLIB, FFS.JOBMACRO, FFS.PROCLIB, PTF.JOBLIB, PTF.JOBMACRO, TESTERL, DUMMY.FILE, DUMMY.FILEL, DUMMY.FILEX, and DUMMY.FILES. The last four data sets are required as "default" data sets by the NIPS procedures. DUMMY.FILES and DUMMY.FILEX should be cataloged to the NIPS system pack since they are never actually called for by the system.

The second function is to restore the system data sets required by NIPS 360 FFS to disk. If the system was received as a dump/restore tape, reload the system data sets to a direct access device by using the IEHDASDR Utility program. If the system data sets were received as unloaded partitioned and sequential data sets, restore them to disk

by first allocating space for them and then reloading them from tape.

The third function in the initialization of the NIPS 360 FFS is putting the procedures into the SYS1.PROCLIB. The procedures are sent as members of the partitioned data set named FFS.PROCLIB. The procedures can be punched from this data set with the Utility Program IEBPTPCH. They can then be modified if necessary or placed directly into the SYS1.PROCLIB with the IEBUPDTE Utility program. If punched procedures are not desired, the "REPRO" capability of the IEBUPDTE program can be used to update the SYS1.PROCLIB.

The final function is to run the sample jobs to test the system installation. The sample jobs are on a sequential data set named NIPS.SAMPLE.JOB as unblocked card images. The sample jobs can be punched and submitted via the card reader or copied to tape and loaded directly from the operator's console (SYSIN tape). If an installation requires specific accounting information, the job cards for all of the sample jobs will have to be changed. (See appendix B.)

Section 5

UPDATES

NIPS 360 FFS will be maintained at a user's installation by means of version updates and Program Transmittal Fixes (PTF). Version updates will be done infrequently and correspond to a major system change.

5.1 Version Changes

NIPS 360 FFS received in an installation package will have a 4-digit version number associated with it. This version number corresponds to a major system upgrade and is the version number referred to in a Discrepancy Change Report (DCR). A version update will usually involve an upgrade of the load modules for an entire component(s) and will not occur very often. It will require replacing the FFS.JOBLIB and/or FFS.JOBMACRO system data sets. Version updates will include all prior PTFs; therefore, system data sets PTF.JOBLIB and PTF.JOBMACRO will be replaced with new data sets initialized for future PTFs.

5.2 Program Transmittal Fix (PTF)

NIPS 360 FFS may be updated by applying PTFs to a given version of the system. PTFs will be received by an installation as one or more jobs that are processed by the Operating System like any other batch job. When executed, these PTF's will perform the update of the specific part of NIPS 360 FFS that is to be upgraded.

Each PTF will have a 2-digit number associated with it. The job card of each PTF will have a job name in the form -

```
//Vnnnnppt JOB  ,,MSGLEVEL=1
```

where

nnnn	=	OS release number/NIPS version number
pp	=	Two-digit PTF number
t	=	S, update to Program Load Library (FFS.JOBLIB) M, update to Macro Library (FFS.JOBMACRO) P, update to NIPS procedures (SYS1.PROCLIB)

A PTF number will be written across the top of each PTF job deck. PTFs which update the Program Load Library will specify a procedure named XFFSPTFL on the EXEC card. This procedure has been moved onto SYS1.PROCLIB along with the other NIPS procedures.

Note: Updates to FFS.JOBLIB and FFS.JOBMACRO will be placed on PTF.JOBLIB and PTF.JOBMACRO respectively.

Section 6

DISCREPANCIES AND CHANGES

NIPS 360 FFS is considered to be a dynamic system which is constantly being improved and expanded. As such, it is unlikely that it will ever reach a final version. This section describes how system users should notify the Command and Control Technical Center (CCTC) of system discrepancies and how they can suggest changes and/or additional capabilities.

The vehicle for reporting a discrepancy or change is the Discrepancy/Change Report (DCR). A copy of the DCR is included in appendix E. This DCR should be sent to:

Director
Command and Control Technical Center
The Pentagon, Room BE685
ATTN: C333 NIPS Project Officer
Washington, D.C. 20301

When a discrepancy is suspected in the NIPS 360 FFS, gather all data on the run, and submit it on a DCR. The only required information on a DCR, in addition to the submitter's name and address, is the system version number for the NIPS 360 FFS system in use. The most important portion of the report is the submitter's description of the problem. In this section, the user should describe the problem in as much detail as necessary to assist in effecting a solution. The remainder of the DCR entries are optional depending on the type of discrepancy or change being reported.

As soon as the DCR is received by CCTC a priority is assigned by the NIPS Technical Support Group and the Contracting Officer Representative (COR), and an entry containing pertinent information on the DCR is created in the consolidated file of outstanding DCRs. At predetermined times, all DCRs for a particular component are evaluated by the NIPS Technical Support Group and the COR to prepare a list of DCRs to be scheduled for analysis and resolution.

INSTALLATION

Once a DCR has been activated, a maintenance programmer may request additional information from the submitter. When the problem has been fixed and tested, the solution is recorded on a Maintenance Programmer's Report (see appendix F). At this point, a Program Transmittal Fix (PTF) has been created and may be mailed to the originator depending on the severity of the problem. Applying this PTF to the user's system will resolve the problem until the next scheduled version of NIPS 360 FFS is released.

6.1 Reporting Discrepancies

This discussion of system discrepancies is limited to three types. These are system abnormal termination, failure of the system to function properly, and errors in documentation. These types of errors should be reported to the CCTC so that appropriate action can be taken. It is expected that each user installation will have system maintenance personnel available to distinguish system problems from environmental and utilization problems.

System abnormal termination is the easiest discrepancy for the user to identify since it is usually followed by an ABEND dump. In the procedures included with the installation package, a SYSUDUMP is normally specified. When a dump is taken by the NIPS 360 FFS programs, it should always be included with the DCR unless security considerations do not allow. In this case, the job should be recreated with an unclassified file.

In addition to the ABEND dump, the DCR describing an abnormal termination should include the significant portion of the job's input stream. The complete JCL should be listed and also the data when it is germane to the problem. The listing of the JCL can be obtained via the MSGLEVEL parameter on the JOB card.

Discrepancies of the type where the system fails to function properly but does not end with an abnormal termination are more difficult to diagnose. In these cases, the accurate description of the problem on the DCR is critical. In addition, as much information as possible concerning the problem, should be sent with the DCR. As a minimum, the input, output, and JCL for the job should be

included; when a file is involved in the problem, a listing of the FFT should be sent.

The third type of discrepancy may have nothing to do with the successful operation of the NIPS 360 FFS programs, but errors in the documentation can be considered system discrepancies in the broader sense. In reporting documentation errors, it is important that the document in question be properly identified. The document identification should include the document name, document ID number, volume number, page, and change number, if applicable.

It is expected that most DCRs concerning documentation errors will refer to technical discrepancies or deficiencies instead of typographical errors. The discrepancy should be described in the DCR and a recommended solution to the error should be included.

6.2 Suggesting Changes

The DCR may be used to suggest changes to NIPS 360 FFS as well as report discrepancies. Changes suggested by system users will be considered as to their overall effect upon NIPS 360 FFS. Those changes that enhance the effectiveness of the system and do not conflict with previously planned modifications or the overall system design will be considered for inclusion. Scheduling of changes will be based on the amount of programming effort required and the availability of development funds.

In submitting changes, system users should specify the requirements in as much detail as possible on the DCR. The evaluation of the proposed change by the CCTC will be significantly enhanced by the amount of detail provided.

Section 7

TERMINAL PROCESSING COMPONENT

The Terminal Processing (TP) component is a generalized TP Monitor and Supervisor designed to support many different configurations of remote or locally attached terminals. The component presently supports local 2250, local or remote 2260, remote dial-up (switched network) 1050, and remote 2741 (nonswitched) terminals, and local or remote 3270 terminals. This component will allow remote jobs to be executed from display terminals while batch jobs are being executed.

7.1 TP Space Considerations

TP support is available under the MFT-II and MVT configuration of OS/360 and VS/370. NIPS TP has three major programs - a TP Monitor(TPMON), a TP Supervisor(TPSUP) and a TP Driver program(UTTPDRVR). UTTPDRVR is used to initiate the NIPS TP Monitor and TP Supervisor and have them execute within a single region/partition of core.

The core requirements for a single region TP system equal the sum of the core required for the TP Driver, the TP Monitor, the TP Supervisor, and application programs executed under the control of the TP Supervisor.

The TP Driver program requires 1.44k bytes of core. Core requirements for TP Monitor are discussed in section 7.1.1. Section 7.1.2 discusses TP Supervisor core requirements.

7.1.1 TP Monitor

TPMON size will vary depending on terminal configuration and requirements for remote and/or local support. TPMON size may be estimated for a specific installation using the information listed below. Approximate sizes are as follows:

23K	Basic Monitor
2.4K	Local Graphics Support [subtract 1K if only 2260's] [subtract .3K if only 2250's]
4.1K	Remote Support [subtract 1K if there is only one terminal type]
2.2K	BDAM - Required by Basic Monitor
4.5K	GAM - Required for Local Support
5.6K	BTAM - Required for Remote or 3270 Terminal Support
2.2K	BSAM - Required for recording Accounting Data (optional)

In addition, TPMON requires:

a. 150 bytes for each serviceable terminal

b. 2K for control blocks

c. 1K for each remote communications line.

d. 1K if the full page buffer option is used

e. 3.5K for the TPPAGE program.

f. An additional amount for each terminal if the incore IMQ option has been specified. Eighty bytes per IMQ line is required for each terminal. Thus, if IMQ=25 were specified and the monitor were generated

for two terminals, 4K bytes would be required.

All installations will require the basic Monitor, the TPPAGE program, BDAM access method, and at least part of the local or remote support group together with the necessary access method.

7.1.2 TP Supervisor

The TP Supervisor (TPSUP) requires approximately 10K of core. However, the primary function of TPSUP is to control the execution of the TP applications programs. Therefore, the core requirements of the TP applications programs comprise the most dynamic portion of the total NIPS TP core requirements. Absolute core requirements cannot be defined for the NIPS QUIP and SODA applications. These requirements depend on such factors as:

- The number of terminals in concurrent operation.
- Processing block size. The QUIP process block must be large enough to contain the largest record in the file being processed. The process block size for QUIP is computed from the file's statistics record (N record) to permit processing of the largest data record in the file. An N record is always present for an ISAM file. The default process block size for a SAM file without a statistics record is 10K. The SODA process block size must be large enough to hold the fixed set plus all referenced sets of the data record.
- Buffers. QUIP and SODA both require two I/O buffers. Each buffer must be large enough to contain one block of data (as determined by the data file blocksize).
- The size of user tables and subroutines.
- The size of the user written application (i.e., query or logic statement)

- Concurrent operation of different TP application programs (e.g., QUIP and SODA).

- Core fragmentation and work areas. Free core is not always available in contiguous areas in core. Therefore, a 5K core requirement cannot be satisfied by OS if there are only 4K contiquous areas of free core remaining in the user's region/partition. Core fragmentation will have a greater impact in a multi terminal environment than for a single terminal, since core allocation will tend to be more complex, with multiple applications seeking and freeing core. A limited amount of temporary work space must also be allocated from the free core area.

- Use of the INCORFFT option for the QUIP component. This option allows QUIP to maintain control and FFT information in core storage for files processed while QUIP is in signon mode. Each field/group of an FFT that is maintained in core requires 38 bytes. In addition, the statistics (N) record is also maintained in core. QUIP itself requires an additional 3K bytes for its programs which maintain the FFTs in core.

- The load module structure chosen for the QUIP component. Two load module structures are available for QUIP: TPQUIP and TPQUIPVS. The Monitor and Supervisor are distributed to use the TPQUIP structure. (See Section 7.15, TPQUIPVS Load Structure for QUIP.)

Estimated core requirements for the first seven of the above factors can be obtained by executing a typical application in the associated batch component (i.e., QUIP, FM), and using the option to generate Run Optimization Statistics (ROS). ROS will produce statistics for the following core requirements:

Component
Buffers
Process block
Tables and subroutines
User written applications.

The effect of the above factors on TP core requirements will be discussed first in a single terminal environment, and then in a multiterminal environment.

7.1.2.1 TP Supervisor for Single Terminal

For purposes of illustration, a sample environment will be defined as follows:

- Size of largest data record - 5K
- File blocksize - 7K
- User tables and subroutines - 6K
- Size of user logic statement - 2K
- Size of user query - 3K

The TPSUP core requirements to support a single terminal for QUIP can be estimated as follows:

Supervisor	10K
QUIP component (TPQUIP load structure)	56K
QUIP component (TPQUIPVS load structure)	104k
*Processing block(size of largest record)	5K
*User tables and subroutines	6K
*Buffers (2 buffers x 7K)	14K
*User query	3K
*Total estimated core (TPQUIP load structure)	90K
*Total estimated core (TPQUIPVS load structure)	138K

*Based on sample environment

The core requirements to support a single terminal for SODA can be estimated as follows:

Supervisor	10K
SODA component	40K
*Processing block	5K
*User tables and subroutines	6K
*Buffers (2 buffers x 7K)	14K

INSTALLATION

*Logic Statements	2K
*Total estimated core	73K

*Based on sample environment

The core requirements for EDIT are not affected by the number of terminals or the user application. The estimated core requirements for EDIT are as follows:

Supervisor	10K
EDIT (without VERIFY)	16K
EDIT (with VERIFY)	116K

A user executing both SODA and QUIP from a single terminal can normally plan on running in a region/partition size which supports the larger of these two applications. The user must terminate QUIP before initiating SODA from the same terminal. It is possible however, to initiate QUIP without signing off from SODA. SODA is terminated by issuing an UPDATE or CANCEL command. If the user attempts to issue a QUIP query without properly terminating SODA, the Supervisor will require core for both QUIP and SODA.

A user can initiate either QUIP or SODA from a terminal while being signed on to EDIT. Therefore, the core requirements for EDIT must be added to the core requirements for QUIP or SODA to run EDIT concurrently with those applications.

7.1.2.2 TP Supervisor for Multiple Terminals

The core requirements to support multiple terminals depend on the number of concurrent TP applications which are anticipated. Terminals entering data on the Input Message Queue (IMQ) or paging through data on the Output Message Queue (OMQ) do not affect TPSUP core requirements.

A single copy of QUIP, SODA, or EDIT will support any number of terminals. However, each active QUIP (i.e., in query translation or file search) or SODA (transaction processing) terminal will require additional core for:

Processing block
Buffers
Tables and subroutines
User written application
Data control blocks
File locate and read programs, (nonreentrant)
Internal table and work areas.

As an example, assume the following multiterminal environment:

- Ten terminals
- Three terminals active using QUIP
- Two terminals active using SODA
- Two terminals active using EDIT
- Size of largest data record - 5K
- File blocksize - 7K
- User tables and subroutines - 6K
- Size of user logic statements - 2K
- Size of user queries - 3K

The TPSUP core requirements to support the above sample TP environment can be estimated as follows:

Supervisor	10K	
QUIP component (TPQUIP load structure)	96K	
QUIP component (TPQUIPVS load structure)	178K	
*QUIP processing blocks (3)	15K	
*QUIP tables and subroutines (3)	18K	
*QUIP buffers (6)	42K	
*QUIP user queries (3)	9K	
SODA component	40K	
*SODA processing blocks (2)	10K	
*SODA tables & subroutines (2)	12K	
*SODA buffers (4)	28K	
*SODA logic statements (2)	4K	
EDIT (with VERIFY)	116K**	
*Total estimated core(TPQUIP load structure)		400K
*Total estimated core (TPQUIPVS load structure)		482K

*Based on sample environment

**Without VERIFY - subtract 100K if VERIFY will not be used in EDIT.

7.2 TP Installation

The TP component requires the following additional installation procedures to include it as part of NIPS 360 FFS:

a. Update SYS1.SVCLIB - A Type III SVC module must be added to the SYS1.SVCLIB if NIPS TP is to be run and the operating environment is S/360 or S/370 VS1 or S/370 VS2 Release 1. The TP SVC is not required in an MVS environment. Generation of the TP SVC involves specifying options on the QTPSVCGN macro, assigning a valid Type-III SVC number and executing the TPSVCUPD job. The SVC number chosen has the following implications. The TPMON, as distributed, is generated to use a Type-III SVC number 240. If your SYSGEN defines SVC 240 as a Type-III user SVC and it is not being used, put this number in the space provided for it (IGC00NNN). If your system does not define 240 as a Type-III user SVC, or if this SVC has already been used, check to see if another Type-III SVC number is available. If no Type-III SVC numbers are available, the Operating System must be SYSGEN'ed to provide one. If one exists, place this number in the space provided. If an SVC number other than 240 is used TPMON will have to be modified to reflect this new SVC number (see TP Monitor Generation). After the SVC has been added to the SVC library and before attempting to execute NIPS/TP, the Operating System should be re-IPLed.

   Specification of the TPSVC options is done by using the QTPSVCGN macro in the following format:

```
QTPSVCGN  GAM2250=NO,CP67=NO
                  YES     YES
```

   where:

```
GAM2250=NO
        YES
```

   The GAM2250 option defaults to NO which indicates that no 2250 device will be released by NIPS/TP, during

its execution, to an independent graphics program. YES should be specified only in environments where 2250 devices are supported by TP Monitor, and are going to be referenced by independent graphics packages

CP67=<u>NO</u>
YES

The CP67 option defaults to NO which indicates that the run environment is <u>not</u> a CP67. YES should be specified if the operating environment is a CP67 and the need for DIALing real terminals to virtual port addresses exists.

Error conditions produced during the TPSVC generation are described in section 7.10 of this document.

NOTE: If the TP SVC is required, it is necessary to execute the TPSVCUPD job even if all the default values in the QTPSVCGN macro are taken. This is because two functions required by NIPS/TP must be executed in OS Supervisor state.

The following is a sample deck for generating the TPSVC.

```
//TPSVCUPD      JOB
//               EXEC ASMFCL,PARM.ASM=(LOAD,NODECK),
//                    RM.LKED=MAP
//ASM.SYSLIB    DD
//               DD  DISP=SHR,DSN=PFS.JOBMACRO
//ASM.SYSIN     DD *
               QTPSVCGN
               END
//LKED.SYSLMOD DD DSN=SYS1.SVCLIB(IGC00NNN),DISP=OLD
```

The QTPSVCGN macro statement, with all the default values taken, will result in the TPSVC being generated with the two basic functions.

The last card in the TPSVCUP job applies to the linkedit step of the ASMFCL procedure. The DSNAME parameter on the SYSLMOD card points to SYS1.SVCLIB which is to contain the generated SVC. The value in parenthesis is the SVC name - IGC00NNN. The proper SVC number must be given. It is important to note that Type-III SVC numbers must have a zone (12) punch over the low-order digit. If the SVC number is 240, the low-order digit must be a 12-0 punch.

b. Allocate Input/Output Disk Queue Space -

- Input Queue - TPMON requires an input disk queue data set if the incore IMQ option is not selected (see section 7.4, TP Monitor Generation, paragraph a, Selecting TP Monitor Options).

  This data set is created and cataloged by executing the TPQ job provided in appendix A. This job will provide a queue space sufficient for 50 input lines from each of 10 terminals on a 2314 disk pack labeled FPSLIB.

  If more than 10 terminals are being used at an installation, change the space requirement on the DD1 DD statement for the TPQ job to SPACE=(80,X) where:

  X = 1+(50 * total number of terminals)

  If the TPIMQ data set is not required, i.e., an incore IMQ has been specified, remove the DD1 data definition statement in the TPQ job.

- Output Queue - TPMON requires an output data set for each terminal in the system.

  They must be permanent data sets but not cataloged. All output queue names take the form T.terminalname. The default data set names for the 2250 output queues are T.DD52250A and T.D512250A. Those for local 2260s are called T.DD62260A, T.DD62260B,

...etc. For remote terminals, the terminal name following the 'T.' must be supplied by the user. This name must correspond to the unique terminal name appearing in the polling list (see TP Monitor Generation - QTPLINE macro). The TPQ job provided in appendix A will create two 2250 output data sets and eight local 2260 output data sets on a disk pack labeled PPSLIB. Fewer Output Message Queues can be created by removing the appropriate number of DD statements from the back of the deck. Additional message queues are provided by adding more DD statements to the back of the deck, following the naming conventions described above.

- o EDIT Queue - An alternate output queue is required when the EDIT component is used.

  All attributes of this queue are identical to the Output Queue except the prefix character of the data set name. For EDIT queues, this character is E (i.e., E.DD52250A).

Important: If TPIMQ is allocated, the Output Message Queues must be created on the same disk pack as the Input Message Queue (TPIMQ) since this is assumed in the TP Job Control Language. If an incore IMQ is used, the volume serial number of the disk pack containing the Output Message Queues must be specified on the OMSGQ DD statement in the TP Job Control Language.

Note: Terminals named using the naming option (see TP Monitor Generation) require output queues with corresponding names; e.g., 'T.USERNAME'.

c. Check the TP Monitor - TPMON, as distributed, will support zero, one or two 2250-1 (types 1, 2, or 3) and zero to eight 2260-1s, all locally attached. If the installation has a greater number of either terminal type, but only wishes to assign terminals within the limitations indicated above, the distributed TPMON may still be used. This Monitor will process QUIP (TPQUIP load structure), FMSODA,

DUMP, EDIT, COEDIT, BLAST, ACCESS, RECORD, VIEW, ODE and independent graphics routines.

If TPMON has to be modified for (1) a different SVC number (or no SVC is required), (2) a greater number of terminals, (3) assigning names to terminals, (4) generating remote support, (5) an incore IMQ or (6) use of the TPQUIPVS load structure for QUIP (see Section 7.15, TPQUIPVS Load Structure for QUIP), read the TP Monitor Generation section before proceeding to subparagraph d.

d. Check the TP Supervisor - TPSUP, as distributed, contains a program name table allowing operation of the following problem programs: TPDUMP, TPLIST1, GGINTFAC, TPQUIP (TPQUIP load structure), TPFMSODA, TPEDIT, TPCOEDIT, TPBLAST, TPACCESS, TPVIEW, and TPODE. Three empty (extra) entries are also included in the program name table, for addition of user programs at execution time.

   If TPSUP has to be modified for (1) adding or deleting specific programs from the program entries in the program name table, (2) changing the number of empty entries in the program name table, (3) specifying dynamic mounting of user disk packs, (4) specifying a "lockout" of other program requests when a Sign-on program is signed on, (5) specifying an MVS environment or (6) use of the TPQUIPVS load structure for QUIP (see Section 7.15, TPQUIPVS Load Structure for QUIP), read the TP Supervisor Generation section before proceeding to subparagraph e.

e. Catalog the Default Data File - QUIP will dynamically mount files only so long as the DSNAME field of the DATAFILE DD card (for ISAM files) or SAMFILE DD card (for SAM files) contains the name of a data set of the form DUMMY.name (e.g., DUMMY.ISAMFILE). The data set(s) do not have to exist. They must be cataloged.

f. Start TPMON and TPSUP - The TP component requires two tasks called TPMON and TPSUP. The UTTPDRVR routine is executed to initiate the monitor and

supervisor and to have them execute in a single region or partition.

There is one NIPS 360 FFS procedure for starting the distributed Monitor and Supervisor. A detailed discussion of this procedure is found in the TP Procedures section.

To start TP the following JCL statements are required:

```
//TP JOB  (Optional Data)
//GO EXEC  XTP
```

The single region control program, UTTPDRVR, is executed to attach the TP Monitor and TP Supervisor. UTTPDRVR obtains the PARM parameter information from the EXEC statement to attach the appropriate TPMON and TPSUP load modules. In a S/370 MVS environment, the unit type to be used for dynamic allocation of temporary data sets by application programs may be specified here.

The format of the PARM parameter is:

```
PARM=[,'TPMON    ] [,TPSUP    ]  [,WORKUNIT=SYSALLDA ]
     [ TPMONXXX  ] [ TPSUPXXX ]  [                XXXXXXX ]

                   [,TP=REFORMAT50 ']
                   [    REFORMATnnn ]
```

where:

TPMON
TPMONXXX — The name of the TP Monitor load module. The default is TPMON. The user may specify an alternate load module name. The first five characters of the alternate name must be TPMON, and the monitor must be linkedited on the NIPS system library referenced in the STEPLIB DD statement in the XTP procedure.

TPSUP
TPSUPXXX — The name of the TP Supervisor load module. The default is TPSUP. The user may specify an alternate load

module name. The first five characters of the alternate name must be TPSUP, and the Supervisor must be linkedited on the NIPS system library referenced on the STEPLIB DD statement in the XTP procedure.

WORKUNIT=SYSALLDA
XXXXXXX — The generic name or device type to be used for dynamic allocation of temporary data sets by application programs in a S/370 MVS environment. The default is SYSALLDA.

TP=REFORMAT50
REFORMATnnn — The number of lines that are assigned per terminal on the DASD Input Message Queue. The default is REFORMAT50 which causes 50 lines per terminal to be formatted on the IMQ. The DASD IMQ is reformatted when the monitor is restarted. The user may assign a different number of lines per terminal by specifying REFORMATnnn, where nnn is 1-999, as long as the total SPACE assigned to the IMQ data set will accommodate all the terminals. This parameter has no meaning if an incore IMQ is used.

The UTTPDRVR program loads and attaches the monitor and supervisor load modules. The monitor is attached with the same priority as the control program and the supervisor is attached with a priority of one less than the monitor. A WAIT is issued for the completion of either task. If either task terminates abnormally both tasks are detached and deleted. Both tasks are then restarted.

When the monitor completes its initialization, TP TERMINALS OPEN is displayed on the operator's console. When the supervisor completes initialization, TP SUPERVISOR READY is displayed on the console. Both are nonending tasks and

require manual cancellation. To assist in this, the monitor also types the following request to the operator:

> TP STANDING REQUEST. REPLIES ARE 'ENA', 'DISA', 'MSG', 'PRTY', 'TPS', OR 'TPM'.

This remains as an outstanding request for the duration of the TP job. The reply of 'ENA' is used by the operator to activate a terminal allocated to the TP job (see section 7.14.8 of this document for a description of this option). The reply of 'DISA' is used by the operator to place a terminal allocated to the TP job in an inactive status (see section 7.14.8 of this document for a description of this option). The reply of 'MSG' is used by the operator to send messages to terminals (see section 2.5.2.2 of the Terminal Processing Manual for a description of this option). The reply of 'PRTY' is used to alter the priority of a terminal (see section 7.12.1 of this document for a description of this option). If the operator replies 'TPS', the TP Supervisor task will come to a normal end. If the operator replies 'TPM' both the TP Monitor and TP Supervisor tasks will end. If terminal users are still active when a request has been made to cancel the TP Sueprvisor task, the operator is given the option of restarting the TP Supervisor. Otherwise the TP job will end. The operator may also use the standard OS job cancellation command to terminate the TP job.

7.3 TP Procedure

The XTP cataloged procedure is used to execute the UTTPDRVR routine. The function of this routine is to provide a means to initiate the NIPS TP Monitor and TP Supervisor and have them execute within a single region or partition. The monitor is attached with the same priority as the UTTPDRVR program, and the supervisor is attached with a priority of one less than TPMON.

To provide a better understanding of the procedure and to assist in tailoring the procedure to an installation, each DD statement of a typical single region XTP procedure will be discussed.

INSTALLATION

Execution of UTTPDRVR requires DD cards to support both the TP Monitor and TP Supervisor. To support the TP Monitor, the XTP procedure requires DD cards to identify the NIPS program libraries, two or three DD cards for disk data sets, one or more DD cards for 2260, 2250, 1050, 3270, and/or 2741 terminals, and other optional DD cards. The information required in each DD card is as follows:

```
//STEPLIB DD DSN=PTF.JOBLIB,DISP=SHR
//        DD DSN=FFS.JOBLIB,DISP=SHR
```

These DD statements identify the NIPS program libraries.

```
//INMSGQ   DD   DISP=SHR,DSNAME=TPIMQ
```

This data set was created and cataloged when the TPQ job in appendix A was executed if an incore IMQ was not specified (see section 7.4 TP Monitor Generation paragraph a, Selecting TP Monitor Options). If an incore IMQ is being used, the DD statement is not used.

```
//OUTMSGQ  DD  DISP=SHR,VOLUME=REF=*.INMSGQ
//OMSGQ DD DISP=SHR,VOLUME=REF=INMSGQ
```

These DD statements is used to reference all Output Message Queues when the Input Message Queue is allocated on disk.

```
//OUTMSGQ DD DISP=SHR,VOL=SER=volid,UNIT=unitname
//OMSGQ DD DISP=SHR,VOL=SER=volid,UNIT=unitname
```

These DD statements are used to reference all Output Message Queues when the Input Message Queue is maintained in core. The volid and unitname must be supplied to identify the volume on which all Output Message Queues are allocated.

```
//STATRECS   DD  SYSOUT=A
```

This DD statement defines the data set used by the monitor to record all TP accounting records. It is required if the QTPMOPT macro specified ACCTNG=YES at Monitor generation time. The distributed Monitor requires this DD statement.

```
//ACCTSAVE  DD  TAPE-OR-DISK
```

This DD statement defines the same information as the STATRECS DD statement. Its purpose is to save the accounting information either on tape or disk for further processing. Absence of this card will result in no action being taken to produce the accounting records on some media other than printed output.

```
//DD5   DD  UNIT=XXX
//D51   DD  UNIT=XXX
//DD6   DD  UNIT=(2260-1,8)
```

The above DD statements define the 2250 and local 2260 terminals. In the case of the distributed Monitor, these are the only terminal definition cards necessary. Additional terminal DD cards are added as described in TPMON Generation.

Only the UNIT=operand is required, using any valid name acceptable at the installation. UNIT=hexaddr is always valid for the 2250 and UNIT=(2260-1,n) for the 2260 terminals where n is the number of 2260's desired.

Provide the UNIT hex addresses for the 2250s at your installation for XXX in the DD5 and D51 DD statements. If no 2250 support is required, remove the DD5 and D51 statements from the procedure.

The distributed TPMON is generated to support zero, one, or two 2250 and zero to eight local 2260 terminals. The distributed procedure for executing the TPMON is designed for running with one 2250 and one to eight local 2260s. Consequently, modifications will have to be made to the procedure for the unique terminal configuration of any particular installation.

The difficulty involved in removing from the procedure DD cards defining serviceable units suggests that the easiest method for executing the TP job, at installations with several terminal configurations, would be to add the necessary DD cards as overrides at run time.

If a user wishes, he may create a procedure to be run with any combination of a particular configuration of 2250s,

and local and remote 2260 and 3270 terminals. To do so, in addition to the above, the user should include:

a. One DD card for each 2250 generated into TPMON. The names of these DD cards must correspond to the label appearing in the name field of the QTPDD macro statements for the 2250s (see TP Monitor Generation). These DD statements should take the form:

```
//DD5   DD   UNIT=XXX
```

where XXX is the hex address of the 2250 being allocated.

b. One DD card for each group of one to eight 2260s. As was the case with 2250s, there should be one DD card in the procedure for each QTPDD card on the TPMON generation deck defining 2260s. These DD statements should take one of two forms for 2260 groups [i.e., those defined with UNITS= two or more on the QTPDD card]

```
//DD6  DD  UNIT=(2260-1,N)
```

where N = the number of 2260s to be allocated (may be any nonzero number equal to or less than the number in the operation field of the 'UNITS' operand on the corresponding QTPDD card).

For individually defined 2260s

```
//D61  DD  UNIT=XXX
```

where XXX is the hex address of any particular 2260.

c. One DD card or concatenated string of DD cards for each 1050, 2741, 3270 or remote line. If these terminals have system-generated group names, then only one DD card for each terminal type need be present. If no system-generated group names exist, or if specific telephone lines are desired, then each line should be allocated by hex address. A concatenated string of DDs for each terminal type

is then required with the label in the name field on the QTPDD and QTPLINE macro statements for that terminal type.

These DD statements should take one of the following forms:

```
//RE1  DD  UNIT=(groupname,n)
```

where

groupname = the SYSGEN'ed groupname
n = the number of lines.

```
//RE1  DD  UNIT=XXX
//     DD  UNIT=XXX
//     DD  UNIT=XXX
```

where XXX = the hex address of each line.

d. The desired number of NOUNITn DD cards, the purpose of which is to allow the user to allocate only those units which he specifically wishes to use.

The format of these DD cards should be:

```
//NOUNITn      DD      DDNAME = DUMMY or for symbolic
(or any unique name)             procs default to any
                                 name not in the Monitor
                                 JCL.
```

There should be n-1 NOUNITn DD cards, where n = the number of DD cards in the procedure defining units or groups of units for the TPMON to service.

These DD cards must appear in the procedure before any of the DD cards defining serviceable units.

To prevent any unit or groups from being allocated, a user must:

a. Override the DUMMY with the DD card(s) defining device(s) which are not required.

b. Override the NOUNITn DD card(s) so that the DDNAME parameter specifies the DD name(s) of the unnecessary DD cards.

Note: Simply overriding with DD DUMMY for the unwanted DD cards is not sufficient.

For example, a user whose procedure contains two DD cards for 2250s and one DD card for 2260s may run without allocating any 2250s by coding:

```
//     EXEC    XTP
//NOUNIT1  DD      DDNAME=DD5
//NOUNIT2  DD      DDNAME=D51
//DD5      DD      DUMMY
//D51      DD      DUMMY
```

A procdure for TP may be created which contains all the statements necessary to allocate all of the defined terminal devices. The following TP procedure illustrates all the statements needed for the TPMON. Other DD statements are required to support TP application programs and are described in subsequent paragraphs.

```
//XTP         PROC  TPIMQ=TPIMQ,                                    X
//                  VOMQ='REF=*.INMSGQ',UOMQ=,                      X
//                  UDD5=2E0,UD51=2F1,UDD6='2260-1'                 X
//                  UD61='2260-1',NDD6=8,ND61=5,                    X
//                  URE1=021,URE6=02F,OMIT1=NULL1,                  X
//                  OMIT2=NULL2,OMIT3=NULL3,OMIT4=NULL4,            X
//                  OMIT5=NULL5
//TPMONSUP EXEC PGM=UTTPDRVR,TIME=1439
//INMSGQ       DD       DISP=SHR,DSNAME=&TPIMQ
//OUTMSQ       DD       SPACE=(TRK,0),VOLUME=&VOMQ,UNIT=&UOMQ
//STATRECS     DD       SYSOUT=A
//SYSUDUMP      DD      SYSOUT=A,SPACE=(TRK,(10,50))
//NOUNIT1           DD       DDNAME=&OMIT1
//NOUNIT2           DD       DDNAME=&OMIT2
//NOUNIT3           DD       DDNAME=&OMIT3
//NOUNIT4           DD       DDNAME=&OMIT4
//NOUNIT5           DD       DDNAME=&OMIT5
//DD5          DD       UNIT=&UDD5
//D51          DD       UNIT=&UD51
//DD6          DD       UNIT=(&UDD6,&NDD6)
//D61          DD       UNIT=(&UD61,&ND61)
```

```
//RE1           DD       UNIT=&UPE1
//RE6           DD       UNIT=&UPE6
```

To execute TP and allocate two 2250s, 13 local 2260s, three remote 1050s and two remote 2260s, one need only code

```
//  EXEC   XTP
```

To execute TP without allocating any remote devices, one would code

```
//  EXEC  XTP,OMIT1=RE1,OMIT2=RE6
//RE1       DD      DUMMY
//RE6       DD      DUMMY
```

To execute TP and allocate only one local 2260, one would code

```
//  EXEC  XTP,OMIT1=DD5,OMIT2=D51,OMIT3=DD6,      X
//                OMIT4=RE1,OMIT5=RE6,ND61=1
//DD5  DD   DUMMY
//D51  DD   DUMMY
//DD6  DD   DUMMY
//RE1  DD   DUMMY
//RE6  DD   DUMMY
```

To support the TP Supervisor task the following statements should be added to the TP procedure, depending on the TP application program(s) being run and the OS/VS environment.

```
//SYSUT1  DD  SPACE=(TRK,0),UNIT=2314
//SYSUT2  DD  SPACE=(TRK,0),UNIT=2314
//SYSUT3  DD  SPACE=(TRK,0),UNIT=(2314,SEP=SYSUT1)
//SYSUT4  DD  SPACE=(CYL(0,1)),UNIT=2314
```

The above DD statements define work data sets for QUIP and SODA. They are not required in an MVS environment.

```
//DATAFILE  DD  DISP=SHR,DSNAME=DUMMY.FILE,      X
//                UNIT=(2314,P,DEFER)
```

This DD statement is used to allocate the ISAM data file if it consists of one volume. It is not required in an MVS environment.

```
//SAMFILE  DD  DISP=SHR,DSN=DUMMY.FILES,  X
//              UNIT=(2314,P,DEFER)
```

This DD statement is used to allocate the SAM data file, which must be direct access resident. It is not required in an MVS environment.

```
//FFT DD DISP=SHR,DSNAME=DUMMY.FILE,
//       UNIT=(2314,P,DEFER)
```

This DD statement is used to allocate the ISAM FFT used with a non-NIPS data file allocated by the DATAFILE or SAMFILE DD statement. This statement is not required in an MVS environment.

```
//DATAFIL1  DD  DISP=SHR,DSNAME=DUMMY.FILE,
//               UNIT=(2314,P,DEFER)
//DATAFIL2  DD  DISP=SHR,DSNAME=DUMMY.FILE,
//               UNIT=(2314,P.DEFER)
```

These DD statements are used to allocate the ISAM secondary data files when Interfile Output is used. If more than two secondary files are required, additional DD statements with the names DATAFIL3 through DATAFIL9 are required, one for each additional secondary file. A maximum of nine secondary files is allowed. The parameters required on each of the additional DD statements are the same as those above. These statements are not required in an MVS environment.

```
//FFT1 DD DISP=SHR,DSNAME=DUMMY.FILE,
//       UNIT=(2314,P,DEFER)
//FFT2 DD DISP=SHR,DSNAME=DUMMY.FILE,
//       UNIT=(2314,P.DEFER)
```

These DD statements are used to allocate the ISAM FFTs used with non-NIPS data files allocated to the DATAFIL1 and DATAFIL2 DD statements. If more than two secondary files are required, additional DD statements (named FFT3 through FFT9) are required, one for each additional secondary file. A maximum of 9 secondary file FFT references is allowed and

the parameters for each additional DD statement are the same as above.

If System Management Facilities (SMF) information is required for secondary files, DUMMY.FILE should be replaced by the secondary file name. There should then be a unique DATAFILn DD statement for each secondary file that is to be referenced by QUIP.

In an non-MVS environment, QUIP and SODA are able to dynamically allocate files only so long as the DSNAME fields on the DATAFILE and DATAFILn DD statements reference the name of a dummy file (i.e., a DSN beginning with "DUMMY."). If DATAFILE points to any other data file, then only files on the same pack may be queried as the primary file. Similarly, if any DATAFILn points to a specific file, then only files on that same pack may be queried as secondary files associated with that DD statement. Attempts to query files on other packs will result in an S413 ABEND. In an MVS environment dynamic allocation of the requested file is performed without the requirement for a DD statement.

```
//DATAFILA DD UNIT=(2314,2),DSN=DUMMY.FILA,
//             SPACE=(TRK,0)
```

This DD statement is used to allocate the data file if it requires two volumes. It is not required in an MVS environment.

```
//DATASMFx DD DSN=filename,DISP=SHR
```

This DD statement defines a data file for which System Management Facilities (SMF) information is required. The suffix on the ddname may be any unique numeric/alpha character. The 'filename' is the data set name for a file to be queried. One DD statement is required for each file that is to be queried by QUIP. It is not required in an MVS environment.

```
//SLIB DD DISP=SHR,DSN=DUMMY.FILEL
//      DD DISP=SHR,DSN=FFS.JOBLIB
```

These DD statements are used to allocate and define the user and system libraries required for the various applications in a non-MVS environment. The user library may

contain (a) tables and subroutines associated with field conversion; (b) QUIP queries and structured RITs to be referenced with the QUIP LOAD operator; (c) formats utilized by FORMATTER; and (e) various EDIT applications. The system library (FFS.JOBLIB) contains standard data conversion subroutines and standard formats for FORMATTER. In an non-MVS environment, a user library may be dynamically allocated only so long as the DSNAME on the first SLIB DD statement references the name of a dummy library (a data set name beginning with "DUMMY."). When this is the case, FORMATTER and EDIT attempt to allocate the specified user library and replce the dummy library with the user library. QUIP does the same for any library specified on a LOAD operator. QUIP and SODA always attempt to allocate a file library. The name of the file library is obtained by adding the suffix "L" to the data set name of the file. (For QUIP, the data set name of the primary file is used if Interfile Output is requested; and if the file, is a SAM file, the ending "S" is replaced by "L" to obtain the library data set name.) If a file library with this data set name cannot be located and the name is a qualified data set name, one final attempt is made to allocate a library using the NIPS file name rather than the full qualified data set name. If the file library can be located and allocated, it internally replaces the dummy library in the concatenation of data sets associated with the SLIB DD statements so that searches for tables and subroutines start with the file library. If the file library cannot be allocated and if a file library is required for successful execution of the application, an appropriate error condition will be displayed.

In an MVS environment these DD statements define the libraries required for any application which does not require the dynamic concatenation of a file library with system libraries (defined on MVSLIBnn DD statements described below). If a file library cannot be located or allocated the libraries defined on the SLIB DD statements will be used.

```
//TPDUMP DD SYSOUT=A
```

This DD statement is used for the output from a snapshot of core taken during any abnormal termination of a QUIP query.

```
//EDITDUMP DD SYSOUT=A
```

This DD statement is used for the output from a snapshot of core taken during any abnormal termination of an EDIT function.

```
//SYSUDUMP  DD  SYSOUT=A
```

This DD statement is used to obtain a core dump during abnormal termination of a run.

```
//SNAPSHOT DD SYSOUT=A
```

This DD statement is used for the output from a SNAP dump of core taken during program error recovery for QUIP and FORMATTER.

```
//PL1DUMP DD SYSOUT=A
```

This DD statement is used to obtain a core dump during abnormal termination of SODA.

```
//SDCONSOL  DD  UNIT=SYSDA,SPACE=(2300,(5,,1))
```

This DD statement is used to store the status of each console or terminal for a SODA run.

```
//SDKNSET   DD  SPACE=(CYL,(1,1)),UNIT=SYSOUT,   X
//              DCB=(RECFM=F,BLKSIZE=1004)
```

This DD statement is used to record all key changes made to the file.

```
//SYSLMOD   DD  SPACE=(TRK,(10,,4)),UNIT=SYSDA,  X
//              DCB=FPS.JOBLIB
```

This DD statement is used to store copies of all logic statements used in a SODA update.

```
//SYSPRINT  DD  SYSOUT=A
```

This DD statement is used to print PL1 error messages from SODA and to print the output from the TPDUMP program.

```
//SYSONLIN DD SYSOUT=A
```

This DD statement defines the output device that will receive the Output Message Queue when the dump program (TPDUMP) is called.

```
//AMSGQ DD UNIT=SYSDA,SPACE=(TRK,(0,5))
```

This DD statement defines a message volume for Terminal/Terminal communication. If the IMQ or OMQ is sent to another terminal, it is first copied to a temporary data set on this volume by BLAST. It is not required in an MVS environment.

```
//SYSIN  DD  DUMMY
```

This DD statement defines the input data set for the Supervisor. If there is no input, the statement may remain as above with DUMMY indicating no data is to follow. For the distributed Monitor no input data is required. To expand the Supervisor capabilities to support other applications see the discussion under TPSUP Variations.

```
//EDITLIB DD DSN=DUMMY.FILEL,DISP=SHR
```

or for a MVS system:

```
//EDITLIB DD DUMMY
```

This DD statement is used to allocate the user specified library in EDIT; in FORMATTER it is used to allocate the library which contains new formats. It is not required in an MVS environment. In a MVS environment the EDITLIB DD statement may be coded with DUMMY to indicate that no data is to follow.

```
//EDCONSOL DD SPACE=(TRK,(5,,4)),UNIT=SYSDA
```

This DD statement defines a partitioned data set containing communications records used for various sections of EDIT and COEDIT.

```
//XINDEX  DD DSN=DUMMY.FILEX,DISP=SHR
```

This DD statement is used to allocate the index data set for the data file if Secondary Indexing is used. It is not required in an MVS environment.

```
//SUBFILE DD UNIT=SYSDA,SPACE=(CYL,(0,2,10))
```

This DD statement assigns a system work volume on which a QUIP subfile partitioned data set is allocated when requested by the user. In FORMATTER it is used to allocate the library used for new formats. The secondary quantity, which specifies the amount of additional space to be allocated if required, may be increased according to installation requirements. Similarly, the number of directory records may also be increased as required. This statement is not required in an MVS environment.

```
//MENUSET DD DSN=DUMMY.FILEL,DISP=SHR
```

This DD statement is used to define the distribution data set for VIEW. The distribution data set is a partitioned data set; its logical record length must be 80, and its blocksize may be any multiple thereof. The distribution data set contains the previously stored output reports, which can be selected by VIEW. It must contain a member named MENULIST which is made up of titles and descriptions of the stored reports.

Note: VIEW is able to dynamically mount distribution data sets as long as the DSNAME field on the MENUSET DD statement contains a data set name that begins with "DUMMY."

```
//MVSLIB01 DD DSN=FFS.JOBLIB,DISP=SHR
```

This DD statement identifies the first library that is to be concatenated with the file library for field conversion with QUIP and SODA applications in an MVS environment. It is not required in any other environment. Additional libraries that are to be included in the concatenation with the file library can be specified on additional DD statements, each with a unique ddname MVSLIBnn, where nn is a 2-digit sequence number starting with 02 and incremented by one for each data set (up to 15). The order of concatenation is in the MVSLIBnn ascending sequence (e.g., MVSLIB01,MVSLIB02,MVSLIB03,etc.). For those applications which do not employ a file library, this concatenation technique is not applicable. Field conversion, if required, is performed using the libraries identified with the SLIB DD statement.

7.3.1 JCL Considerations for EDIT Generated Batch Jobs

When a SUBMIT command is issued to enter a job stream generated at a terminal for batch processing, EDIT copies the job stream from the terminal's work area to one of three data sets or devices. The selection of a data set and the subsequent method of handling the job stream is dependent on the presence of either a HASPRDR or SHARDASD DD card or the absence of both. EDIT scans the DD names and uses the first applicable DD name found.

If neither DD name is found, EDIT dynamically names, allocates and catalogs a data set for the job stream and activates an internal reader to read it into the system job queue. The name of this reader is TPRDR080. It is distributed with the other NIPS procedures on FPS.PROCLIB and must be copied to SYS1.PROCLIB.

The HASPRDR DD card is included in the job stream to indicate that an internal reader is to be used for job entry. A HASP pseudodevice or a MVS internal reader may be specified depending on the operating system in use.

In a HASP environment the job is submitted to the HASP internal reader. The DD statement must contain UNIT=2520 and DISP=OLD. The 2520 unit is a pseudodevice and must be defined at S/360 OS System generation time. On some HASP systems UNIT=INTRDR should be coded to invoke the HASP internal reader.

In the MVS environment the DD statement must contain SYSOUT=(A,INTRDR). INTRDR is the name of the MVS internal reader. This DD statement is required for the EDIT SUBMIT command to function in a MVS environment.

If the SHARDASD DD card is used, it must contain a DSNAME of user choice, VOL=SER=(serial number) if not cataloged, and DISP=(MOD,KEEP). EDIT assumes the device is a shared DASD and issues the RESERVE macro to lock out other users or computer systems until the job stream is completely written, at which time a DEQ macro is issued to free the device. The job stream is written sequentially in 800-character blocks of 80-character records. The mention of the SHARDASD DD card is not to point out any available option or alternate method of processing these batch jobs.

This DD card merely provides the means for devising other methods and establishing procedures tailored to an installation's requirements. Jobs submitted to this data set can be processed at a later time or concurrently on another computer system if the device is a Shared DASD. In either case, the batch jobs will not compete with online operations for computer resources.

7.3.2 JCL Considerations for ALC Source Code Verification

When an installation will be using the ALC source code verification feature of EDIT, the following DD statements must be included in the TP JCL deck.

```
//ASMIN DD UNIT=&STG,SPACE=(CYL,(0,1))
//ASMOUT DD UNIT=&STG,SPACE=(CYL,(0,1))
//ASMWK1 DD UNIT=&STG,SPACE=(CYL,(3,1))
//ASMWK2 DD UNIT=&STG,SPACE=(CYL,(3,1))
//ASMWK3 DD UNIT=&STG,SPACE=(CYL,(3,1))
```

The above data sets are used to invoke the assembler.

```
//ASMPRT DD SYSOUT=A
```

This DD statement is used when the user invokes the assembler LIST option.

```
//ASMDUM DD DUMMY
```

This DD statement is used to supress undesired assembler output functions.

```
//SYSLIB DD DSN=&JCBMACRO,DISP=SHR
//       DD DSN=SYS1.MACLIB,DISP=SHR
```

The above DD statements provide the assembler with needed macro libraries.

When these DD statements are not included in the JCL deck, EDIT will suppress all attempts to verify ALC source code.

7.4 TP Monitor Generation

Regenerating TPMON involves changing macro statements in the TPMONGEN job and executing it. The macros QTPMOPT, QTPDD, QTPPROG1, QTPPROG2, QTPPROG3, QTPGEND, and QTPLINE may be used to create tables and control blocks used by the monitor. All TP modules, except for the optional installation written validation subroutine, necessary for generating any TPMON are contained in FFS.JOBLIB. (For additional information on the optional installation validation subroutine, see section 7.14 of this manual.) Error conditions produced during TP Monitor generation are described in section 7.10 of this document.

a. <u>Specifying TP Monitor Options</u> - Specify the TP Monitor options using the QTPMOPT macro in the following format:

```
            ┌         ┌     ┐  ┌             ┌     ┐ ┐  ┌                  ┐
QTPMOPT     │ ACCTNG= │ YES │  │ ,FULLPAGE=  │ YES │ │  │ ,VALID=XXXXXXX   │
            │         │ NO  │  │             │ NO  │ │  └                  ┘
            │ ,DVM=   │ YES │ ││ ,IMQ=  │ nn   │ ││ ,ODEWK= │ NO  │
            │         │ NO  │ ││        │ DASD │ ││         │ YES │
            └                                                     ┘
```

The options are keywords as follows:

ACCTNG= YES / NO	The ACCTNG option defaults to YES which indicates that accounting data should be supplied by the TP Monitor.
FULPAGE= <u>NO</u> / YES	The FULPAGE option defaults to NO which indicates single line writes are made from the OMQ to the terminal display screen. If FULPAGE=YES is specified, a buffer large enough for a full page write is created, and writing a full screen to local CTR terminals is accomplished by a single write

instruction. If FULPAGE option is specified as YES at monitor generation time and the type of CRT is a 3277/3275, TPPAGE1 will operate in a formatted mode. TPPAGE1 will then support the 3277 light pen, function keyw and 3284/3286 printers. The 24th and last line on the CRT will be as follows:

o__oNP/1__oNL/2__oPP/3__oPL/4__oS/5__oH/6 oE/7__oP/8

The cursor will be placed in the 2nd position of the line to allow keyboard entry of commands. The next eight fields of the last line are selector pen detectable, and will cause the appropriate function to take place when selected. The number following the '/' indicates the function key to be depressed to perform the selected function. The last field of the line is 'P/8', if this field is selected with the selector pen or function key 8 is depressed it will cause the current page to be printed on the 3284/3286 printer associated with the 3277.

VALID=xxxxxxxx The VALID option is used to specify the entry point name of the optional installation written validation subroutine. The installation validation subroutine can control the rightful access to the TP terminal devices and NIPS data files. The installation validation subroutine will be linkedited as part of the generated TP Monitor. For a complete description of the optional installation validation subroutine, see section 7.14 of this manual.

DVM= NO / YES

The DVM option defaults to NO which indicates that TP executes in a single virtual machine environment. IF DVM=YES is specified TP assumes it is operating on a CP/67 DVM and will accept CP commands while in the NIPS/TP problem program state.

IMQ= nn / DASD

The IMQ operand defaults to DASD which indicates that the Input Message Queue is a data set on a direct access storage device (DASD). It is a BDAM data set which is always open for TP monitor routines and which is open as a BSAM data set each time an application program is started. An internal ("incore") IMQ may be generated by specifying a numeric value between 10 and 99 as a value for IMQ. The value specified determines the number of incore IMQ lines allocated for each terminal. The IMQ is then maintained as a number of core resident arrays, one for each terminal, each line of which is 80 characters in width. The use of an incore IMQ eliminates the need for I/O buffers and the execution of I/O function that would be required for a disk resident IMQ, thereby improving query response time. See section 7.1.1, TP Monitor, for increased core requirements with an incore IMQ.

ODEWK= NO / YES

The ODWEK option defaults to NO which indicates that no TP workarea for FORMATTER will be generated. If ODEWK=YES is specified a 544 byte TP workarea per terminal is generated. 500 bytes of this area are used for

passing parameter information from TP application programs to FORMATTER, 40 bytes are used for the dynamic specification of format name by TP application programs, and four bytes are reserved for future use.

b. Specifying Priority of Local Terminals - Priority of local terminals is specified by the PRTY operand of the QTPDD macro that defines them. Terminal priority will default to 1 if omitted.

c. Increasing the Number of 2260/2250 Terminals - For more than eight 2260 terminals, a QTPDD card should be added for each eight additional 2260s required. Each QTPDD card must have a 3-character label that corresponds to a DD name in the Monitor execution deck. To add 12 more terminals, for example, two QTPDD cards are needed such as:

```
D61  QTPDD  TYPE=2260,UNITS=8
D62  QTPDD  TYPE=2260,UNITS=4
```

Increasing the number of 2250 terminals - the 2250 terminals supported by the distributed TPMON defined in the generation deck by QTPDD macro in the form:

```
DD5  QTPDD  TYPE=2250
D51  QTPDD  TYPE=2250
```

Each additional 2250 requires a QTPDD card with a unique label (i.e., D52, D53, D54). These labels will be the DD statement names defining the devices when the TP job is executed.

d. Eliminating 2250 and/or 2260 Support - To eliminate 2250 support, remove the "DD5 QTPDD TYPE=2250" card from the TPMONGEN job. To eliminate 2260 support, remove the "DD6 QTPDD TYPE=2260, UNITS=8" card from the TPMONGEN job.

e. Indicating Problem Programs, Conversational Programs, or Independent Graphic Routines Other Than MTO, LIST, or PAGE To Be Run Under TPMON and

TPSUP - Any problem programs other than LIST which are to run under the Supervisor must be included on the QTPPROG1 card. For example:

```
QTPPROG1  QUIP,FMSODA,DUMP
```

This example will enable the Monitor to recognize the ¬Q, ¬F, and ¬D commands for executing QUIP, FMSODA, and the DUMP programs in addition to ¬L for executing the LIST program. The distributed TPMON will recognize the ¬Q command for QUIP, ¬F command for SODA, and the ¬D command for TP dump.

Any Conversational programs other than PAGE1 which are to run under the Monitor must be indicated on a QTPPROG2 card. If PAGE1 is the only conversational program to run under the Monitor, then this card may be omitted from the Monitor generation.

Any independent graphic routines which are to run under TPMON must be included on a QTPPROG3 card. If no independent graphic routines are required, this card may be omitted from the Monitor generation deck. Refer to the "Terminal Processing" manual for a description of independent graphics routines.

Note: When a terminal user requests a program, the requested program name is checked against tables of allowable program names in the following order:

1. Monitor functions (RECORD,TIME,STOP,MTO)
2. Conversational programs
3. Problem programs
4. Independent graphics programs.

It is only necessary for the user to enter sufficient characters for the input name request to identify the program desired. In general, one character will suffice to identify the program. However, assume that a user's system contains two

programs beginning with (for example) the letter 'Q' - a problem program named QUIP and an independent graphics program name QUIK. If the user wishes to request QUIK, he must use all four characters of the name to identify the program. A program request of Q, QU, or QUI would always cause QUIP to be loaded, since the problem program table is searched for a matching entry before the independent graphics program table is searched.

f. Generating a Monitor for BTAM Terminals - To generate a Monitor which will support remote 2260, 2741, 3270, and/or 1050 terminals, several cards need to be added to the Monitor generation deck:

o QTPDD - Additional QTPDD cards are necessary for BTAM terminal support. Its format is described below:

```
ddname QTPDD UNITS=n,LINES=n,PRTY=[n        ],
                                  [(n1,n2..)]

     TYPE=[2260           ],PRINTER=name,
          [2250           ]
          [PRINTER        ]
          [(BTAM[,     ])]
                [,LOCAL]
                [,BSC  ]
                [,ASCII]

     NAMES=([pollid   ],name,[pollid   ],name..)
            [hex addr]       [hex addr ]
```

ddname= A 3-character DD name that corresponds to a DD statement in the JCL which specifies the UCB address. Name must be unique. If QTPDD macro is being used to generate a BTAM type terminal, as opposed to a terminal device supported by GAM, an associated QTPLINE macro with the same name must be a part of the Monitor generation.

UNITS = The UNITS operand indicates the number of Unit Status Tables which are to be generated. The default value is 1.

LINES = The LINES operand indicates the number of line addresses. This operand is designed to be used at installations with two or more 2848s all on separate cables and each cable having a unit address on the same multiplexor channel. The LINES operand is not used with local 3270 support. The default value is 1.

PRTY = A list of priorities corresponding to the local termianls identified in the NAMES parameter. (Priority of BTAM terminals is assigned by QTPLINE macro.) If the PRTY operand is omitted, all local terminals identified will assume default priority of 1. The corresponding terminal priority for an omitted entry within the sublist will assume default value of 1.

TYPE = The type of terminal

```
[2260              ]
[2250              ]
[PRINTER           ]
[(BTAM[,       ])  ]
     [,LOCAL  ]    ]
     [,BSC    ]
     [,ASCII  ]
```

2260 - Local 2260
2250 - Local 2250
PRINTER- 3284/3286 printer devices
BTAM - All terminals that are not supported by GAM; specifically, remote 2260, dialed 1050, nonswitched 2741, local and remote 3277.

= Specifies that no additional line type information is needed.

,LOCAL= Qualifies BTAM terminal type to be a local 3270.

,BSC = Qualifies BTAM terminal type to be a remote 3270 using EBCDIC transmission code.

,ASCII = Qualifies BTAM terminal type to be a remote 3270 using ASCII transmission code.

NAMES=([pollid],name,...)
[hex addr],

Pairs of entries which identify a terminal by pollid or hex address and a unique user-defined terminal name. NAMES= is an optional parameter, since the system will assign a default name in the following format:

```
  XXX          XXXX          X
QTPDD name + QTPDD TYPE= + suffix
```

Example:

```
DD6 QTPDD TYPE=2260,UNITS=2
system generated names -
 DD62260A,DD62260B
```

For local 2260, 2250, and BTAM terminals types of 2741 and local 3270, the hex address is paired with the user-assigned name. The user-asssigned terminal name may be a maximum of eight characters.

Example:

```
DD5 QTPDD TYPE=2250,NAMES=(03F,A2250)
DD2 QTPDD TYPE=2260,UNITS=2,
    NAMES=(020,A2260,021,B2260)
DD3 QTPDD TYPE=(BTAM,LOCAL),
    NAMES=(310,A3270)
```

```
DD4 QTPDD TYPE=(BTAM,),
    NAMES=(046,A2741)
```

For BTAM terminals, other than 2741s and local 3270 devices, a pollid is paired with the user-assigned name.

Example:

```
RE6 QTPDD TYPE=BTAM,NAMES=(A0A0,
    A2260)
RE1 QTPDD TYPE=BTAM,NAMES=(6215,
    FIRST)
RE3 QTPDD TYPE=(BTAM,LOCAL),
    NAMES=(311,A3270)
R32 QTPDD TYPE=(BTAM,BSC),UNITS=3,
    NAMES=(4040,A3220,40C2,B3270,
    40C3,C3270)
R33 QTPDD TYPE=(BTAM,ASCII,UNIT=4,
      NAMES=(2020,B3270,2041,B3270,
            2042,C3270,2043,D3270)
```

PRINTER= Used to associate 3284/3286 printers with the 3277 terminal display defined in this QTPDD macro. The name specified is the user-assigned terminal name given in the QTPDD macro which defines the printer. For local printers, a separate QTPDD macro is used to define the printer device; for remote printers, the device is named in the NAMES entry, and is specified as the associated printer via the PRINTER entry. For example:

```
P32 QTPDD TYPE=PRINTER,NAMES=(311,P3270)
L32 QTPDD TYPE=(BTAM,LOCAL),NAMES=(310
      L3270),PRINTER=P3270
R25 QTPDD TYPE=(BTAM,BSC),NAMES=(4040,
      R13270,40C1,R23270,40C2,
      P33270),PRINTER=P33270
```

Examples:

Local 2260 Terminals:

```
D20  QTPDD PRTY=4,TYPE=2260
D21  QTPDD UNITS=3,TYPE=2260,NAMES=(021,
           A2260,022,B2260,023,C2260)
```

Local 2250 Terminals:

```
D3F  QTPDD TYPE=2250
D5F  QTPDD TYPE=2250,NAMES=(05F,A2250)
```

Remote 2260 Terminals

```
RE6  QTPDD UNITS=2,TYPE=BTAM,NAMES=(A0A0,
           FIRST,A0A1,SECOND)
RE7  QTPDD TYPE=(BTAM,),NAMES=(A0A0,
      FIRST)
```

1050 Terminals

```
RE1  QTPDD UNITS=3,TYPE=BTAM
RE2  QTPDD UNITS=2,TYPE=(BTAM,)
```

2741 Terminals:

```
D47  QTPDD TYPE=BTAM,NAMES=(047,A2741)
D48  QTPDD TYPE=(BTAM,)
```

Local 3270 Terminals:

```
L32  QTPDD TYPE=(BTAM,LOCAL),NAMES=(410
      A3270),PRINTER=P3270
L33  QTPDD TYPE=PRINTER,NAMES=(411,
      P3270),UNITS=1
L37  QTPDD TYPE=(BTAM,LOCAL),NAMES=(213
      A3270,214,B3270,215,C3270),UNITS=3
```

Remote 3270 Terminals:

```
R32  QTPDD UNITS=2,TYPE=(BTAM,BSC),
      NAMES=(4040,A3270,40C3,B3270)
R33  QTPDD UNITS=3,TYPE=(BTAM,BSC),
```

```
                    NAMES=(4040,A3270,40C3,B3270,40C4,
                      P3270),PRINTER=P3270
          R34 QTPDD TYPE=(BTAM,ASCII),UNIT=4,
                    NAMES=(2020,A3270,2041,B3270,
                      2042,C3270,2043,D3270)
```

o QTPLINE - There should be a QTPLINE card for all BTAM supported devices. The QTPDD card and associated QTPLINE card must have the same 3-character label in the name field. This label must consequently appear as the DD name for the DD card defining remote units in the monitor JCL. (See section on TP Procedures.) The format is described below.

```
ddname QTPLINE ([pollid],name[,pollid],name...),

     UCB = hex addr, PRTY=[ n          ],
                          [ (n1,n2,...) ]
     TYPE=(device type[,mode][,keyboard arrangement]
                             [,printer device      ]
     [,BSC transmission code])
```

ddname = Must be same 3-character name as appears in QTPDD statement which identified the unit.

([,pollid],name[,pollid],...)= Pairs of entries which identify by polling ID and unique user-defined terminal name. This must be the first operand following the QTPLINE operation code. The terminal name is 1-8 characters.

For 2260s, the polling ID consists of a 4-hex digit; the last two digits will be A0, A1, A2, ... etc.

Example: (A0A0,FIRST,A0A1,SECOND)

For 1050's the polling ID consists of a 4-hex digit; the

last two digits will be 15. Since only one 1050 may use the line at a time, several terminals may share the same line ID.

Example: (6215,FIRST,6215,SECOND,
6215,THIRD)

For 2741 terminals, the polling ID may be omitted since only one terminal may be attached to a communication line.

Example: (,A2741)

For local 3270 terminals, the polling ID may be omitted, since a relative line number for each 3277 associated with the QTPLINE macro will be automatically generated.

Example: (,A3270,,B3270,,C3270)

For remote 3270 terminals, the polling ID consists of 2 hexadecimal digits in EBCDIC or ASCII to represent the control unit and device addresses.

Example: EBCDIC
(4040,A3270,40C2,B3270,40C3,
C3270)
ASCII
(2020,A3270,2041,B3270,2042,C3270)

UCB=hex addr — Hex address which identifies the specific line. May be a list of addresses for the actual lines that the polling list is to be used for. For local 3270s, the hex address must contain the address of the first local 3277 assigned to the line.

PRTY= n
(n1,n2..)

The list of priorities corresponding to the terminals identifies in the poll list parameter. If the PRTY operand is omitted, all terminals identified will assume default priority of 1. The corresponding terminal priority for an omitted entry within the subset will assume default value of 1. Three terminals with respective priorities of 5, 1, and 6 could be coded as:

PRTY=(5,,6) or PRTY=(5,1,6)

TYPE=(device type[,mode][,keyboard arrangement]
[,printer device]

Note: The subparameter for the TYPE keyword are positional in nature. The absence of a positional parameter is indicated by a comma coded in its place.

[,BSC transmission code])

Device type identifies a particular type of device. Possible entries are

[2260
1050
2741
3270]

Mode identifies the mode in which the device is being operated. Possible entries are:

[DIAL]	applies to 1050 devices only
[LOCAL]	applies to 3270 devices only
[REMOTE]	applies to 3270 devices only

Keyboard arrangement identifies keyboard arrangement for the designated 2741 device. Possible entires are:

[S]	Standard IBM Selectric typewriter
[B]	PTTC/BCD code compatibility
[E]	PTTC/EBCD code compatibility

Printer device, if present, identifies the existence of a 3284/3286 printer device on the line, and causes the inclusion of the printer I/O appendage and entry point. The format of the entry is:

[PRINTER]

BSC transmission code identifies the remote 3270 transmission code. Possible entries are:

ASCII
EBCDIC

Examples

2260 Terminals:

```
RE6 QTPLINE (A0A0,FIRST,A0A1,SECOND),
   UCB=02F,PRTY=(5,4),TYPE=2260

RE7 QTPLINE (A0A0,FIRST),UCB=02F,
   TYPE=2260
```

1050 Terminals:

```
RE1 QTPLINE (6215,FIRST,6215,SECOND,
   6215,THIRD),UCB=021,TYPE=(1050,DIAL)

RE QTPLINE (6215,FIRST,6215,SECOND),
   UCB=021,PRTY=(,4),TYPE=(1050,DIAL)
```

2741 Terminals:

```
D47 QTPLINE (,A2741),UCB=047,PRTY=5,
     TYPE=(2741,,S)
D48 QTPLINE(,A2741),UCB=048,
     TYPE=(2741,,E)
```

3270 Terminals-Local:

```
L32 QTPLINE (,A3270),UCB=410,PRTY=4,
     TYPE=(3270,LOCAL)
L33 QTPLINE (,P3270),UCB=411,TYPE=(3270,
     LOCAL,PRINTER)
L37 QPTLINE (,A3270,,B3270,,C3270),
     UCB=213,TYPE=(3270,LOCAL)
```

3270 Terminals-Remote:

```
R32 QTPLINE (4040,A3270,40C3,B3270),
     UCB=025,PRTY=4,TYPE=(3270,REMOTE,,
     EBCDIC)
R33 QTPLINE (4040,A3270,40C3,B3270,40C4,
     P3270),UCB=025,TYPE=(3270,REMOTE,
     PRINTER,EBCDIC)
```

QTPPROG1 name,name,name,...

This macro is used to identify all the problem programs which are to be run under TP Supervisor. The name operand is any TP problem program name other than LIST which is run under the Supervisor, so that Monitor will recognize control commands for executing the program. Names must be six characters or less.

Possible entries are:

```
QUIP
QUIPVS
FMSODA
EDIT
DUMP
```

```
VIEW
ODE
BLAST
ACCESS
RECORD
X$$$
```

NOTE: MTO STOP and TIME are Monitor functions and, therefore, not listed under problem program definition. Either QUIP or QUIPVS may be used to identify the QUIP component. QUIPVS indicates that the QUIPVS load structure is to be used for QUIP applications (see section 7.15 TPQUIPVS Load Structure for QUIP). If TPQUIPVS is specified as the program name on a QTPSPGM macro in the TP Supervisor generation, then QUIPVS must be specified here on the QTPPROG1 macro. If TPQUIP is specified in the TP Supervisor generation, then QUIP must be specified here on the QTPPROG1 macro.

QTPPROG2 name,name,.....

The QTPPROG2 macro specifies the names of conversational programs other than PAGE1 which are to run under the Monitor. If PAGE1 is the only conversational program to run under the Monitor, then this macro may be omitted from the Monitor generation.

QTPPROG3 name,name,...

The QTPPROG3 macro specifies the name of independent graphic routines which are to run under the Monitor. If no independent graphic routines are required, the macro may be omitted from the Monitor generation.

g. Assigning Type III SVC Number Other than SVC 240 - Change the 3-digit SVC number after the equals sign in the QTPGEND macro which has the form:

```
QTPGEND  SVC=240
```

This is not a zoned SVC number. Do not change the order of the cards in the TPMONGEN deck. The QTPGEND macro must be the card before the END TPMCTRL card. The SVC parameter is not required in an MVS environment.

The following is a sample deck for generating a Monitor.

```
//TPMONGEN   JOB
//            EXEC ASMFCL,PARM.ASM=(LOAD,NODECK),PARM.LKED=MAP
//ASM.SYSLIB DD
//            DD  DISP=SHR,DSNAME=FPS.JOBMACRO
//ASM.SYSIN  DD     *
            QTPMOPT ACCTNG=YES
DD5         QTPDD  TYPE=2250
D51         QTPDD  TYPE=2250
DD6         QTPDD  TYPE=2260,UNITS=8,NAMES=(130,LEFT2260,131, X
                   RGTH2260)
D61         QTPDD  TYPE=2260,UNITS=5
RE1         QTPDD  UNITS=3
RE6         QTPDD  UNITS=2
R025        QTPDD UNITS=2,NAMES=(4040,TERM0000,
                  40C3,PRINT000),TYPE=(BTAM,BSC),
                  PRINTER=(PRINT000)
RE1         QTPLINE (6215,A1050,B1050,C1050),TYPE=(1050,DIAL),
                    UCB=021
RE6         QTPLINE (A0A0,A2260,A0A1,B2260),TYPE=2260,
                     UCB=02F
R025        QTPLINE (4040,TERM0000,40C3,PRINT000),
                    TYPE=(3270,REMOTE,PRINTER,
                    EBCDIC),UCB=025
            QTPPROG1    QUIP,FMSODA,A$$$$
            QTPPROG3    GGINTFAC
            QTPGEND     SVC=240
            END

/*
//LKED.SYSLMOD  DD  DISP=OLD,DSNAME=FPS.JOBLIB(TPMON)
//LKED.SYSLIB   DD  DISP=SHR,DSNAME=SYS1.TELCMLIB
```

```
//                DD  DISP=SHR,DSNAME=FFS.JOBLIB
//LKED.SYSIN DD *
     ENTRY TPMCTRL
     NAME TPMON(R)
```

The first three cards in this example need never be altered.

The fourth card is simply the second in a string of concatenated DD cards used for overriding the SYSLIB DD card in the assembly step. The DSNAME parameter should point to the system data set containing the macros necessary for Monitor generation.

The QTPMOPT card is used to specify to the TP Monitor that installation accounting data is to be provided. For further information on this data, see the section - TP Accounting Data.

Cards DD5, D51, DD6, and D61, each containing the QTPDD macro, are used to specify that this Monitor is being generated to support two 2250s and 13 local 2260's. Since the PRTY operand was not specified on the statements, all terminals assume the default priority of 1. The names DD5, DD51, DD6, and D61 must appear as DD names for the specified units in the Monitor execution deck.

The last operand on the QTPDD card - DD6 - is used to assign names to the hex addresses of two of the local 2260s. The other local 2260s will receive their names through the default option (see TPSUP Variations). This operand takes the form:

```
NAMES=(XXX,USERNAME,XXX,USERNAM...)
```

where XXX is the hex address for a 2250 or local 2260. Each hex address is followed by the name to be assigned to that device. When making use of this option, the user must be certain that for each named unit there exists a corresponding output queue named 'T.terminalname'.

Refer back to the TP Installation section for a discussion on TERMINALNAME.

INSTALLATION

The QTPDD cards, RE1 and RE6, together with their corresponding QTPLINE cards are used to indicate remote support consisting of three 1050s and two 2260s. Since the PRTY operand was not coded on the QTPLINE cards, all generated remote 1050s and 2260s assume the default priority of 1. The names RE1 and RE6 must appear as the DD names for the DD cards allocating the remote units in the Monitor execution deck. (See section on TP Procedures.)

The QTPDD card R025, together with the corresponding R025 QTPLINE card is used to show how 3284/3286 printers are defined, and then associated with 3277 terminals. The printer(s) defined by the QTPDD macro is associated to the terminal defined by the QTPDD macro, by the parameter(s) of the PRINTER operand. The QTPLINE macro includes the I/O appendages, TPBT327R and TPBT328R, and their entry points to be used by the line. The printers and terminals will be defined through a DD card which associates the label R025 to the label on the QTPDD and QTPLINE macros.

The QTPPROG1 card lists the programs (other than MTO and LIST which are automatically included) that are to run under the Supervisor. The A$$$$ is used to illustrate that TPMON can be generated with space names for unused characters such as ¬A.

These names take the form A$$$$, Z$$$$. The TPSUP, which actually loads the problem programs and graphic jobs from the library, can be given the correct name at execution time on its SYSIN parameter cards (see TPSUP Variations). For example:

```
PROGLIST=USE TPANAME TPQUIP TPLIST1 TPPMSODA
         GGINTFAC.
```

would supply a library name of TPANAME to use instead of TPA$$$$, whenever ¬A was entered.

The other names are required, since the USE card must identify all programs executable within a given TPSUP region.

The only conversational program to run under this monitor is TPPAGE. Since this is automatically included, no QTPPROG2 card is used.

The QTPPROG3 card is included, however, since the independent graphic routine GGINTFAC is to be run under the generated Monitor.

The QTPGEND card assigns the number 240 to the Type III SVC needed for the Monitor. This must be the last card before the END card.

The last card in the input stream is the END card which indicates the end of assembly.

The last four JCL cards in the deck are concerned with the link edit step of Monitor generation. These cards should appear in the order of the DD statements in the ASMFC procedure. The DSNAME parameter on the SYSLMOD card points to FFS.JOBLIB which is to contain the generated TPMON. The DSNAME parameter on the SYSLIB card points to SYS1.TELCMLIB; which contains the LOPEN routines for 3270 devices and FFS.JOBLIB which contains the component TP modules for the monitor. The SYSIN statement indicates that the linkage editor control cards follow in the input stream.

7.5 TP Supervisor Generation

Generation of TPSUP involves specifications of macro statements in the TPSUPGEN job, then executing it. Two macros, QTPSPGM and QTPSOPT, may be used to create the program name table and specify the control options used by the Supervisor. The program name table is a list of the programs which can be executed by the TP Supervisor, and the attributes of each. The generated program name table can be modified at TPSUP execution time by using the PROGLIST statement, described in section 7.6, TP Supervisor Variations. The FFS.JOBLIB contains all programs necessary to regenerate a TPSUP. Error conditions produced during TP Supervisor generation are described in section 7.10 of this document.

a. Build the program name table, using the QTPSPGM macro. One QTPSPGM macro is required for each program other than the TPLIST1, TPDUMP, GGINTFAC programs (which are automatically generated into the name table with the first occurrence of a QTPSPGM macro). The format is:

QTPSPGM progname,(attrib1,attrib2,.....)[,SFDSN={TERM / TEMP}]

The program name is required as the first operand. For the QUIP component, either TPQUIP or TPQUIPVS may be specified (see section 7.15, TPQUIPVS Load Structure for QUIP). However, the name must correspond to the one specified on the QTPPROG1 macro in the TP Monitor generation. If QUIPVS is specified on the QTPPROG1 macro in the TP Monitor generation, TPQUIPVS must be the program name on the QTPSPGM macro for the QUIP component. If QUIP is specifed on the QTPPROG1 macro in the TP Monitor generation, TPQUIP must be the program name on the QTPSPGM macro for the QUIP component. The second operand is coded within parentheses and specifies attributes from the following list:

OMQ - the output message queue data set is used by this program

INIT - an initializing program is required (the name of the initializer is TIxxxxxx where TPxxxxxx is the name of the program)

SIGN1 - the program is a Sign-on program and only one copy of the program will reside in core no matter how many terminals request the program.

SIGNMLT - the program is a Sign-on program and a copy of the program will be loaded for each terminal request

INDGR - the program is an independent graphic routine

INCORFFT - Control record and File Format Table (FFT) information is maintained in core for the active file when QUIP is in signon mode. This option is valid only for the QUIP component.

When this option is specified, faster QUIP response time is achieved for files processed in signon mode:

1. Control record and FFT information are saved in core so that these file records are usually processed only once, at the time the file is signed on, rather than during the translation of each query.

2. Field label and/or edit mask information is not maintained in core but is obtained from the FFT as required for each query. This prevents possible waste of large amounts of core storage since label and edit mask data is not required for the majority of QUIP operators.

3. When a new file is interrogated by a user running in sign-on mode, required information from the file control records and the fixed information, i.e., exclusive of label and edit mask, from the File Format Table records is saved in an area which is made available to all potential QUIP users of the file. This information is available as long as there is an active user for that file in the TP system.

4. If the user is signed on to another file at the time the new file is referenced, the FFT and control tables for that former file are deleted, providing no other QUIP users are signed on to the file. If

other QUIP users are signed on to the file, the tables are retained for their use.

5. When the user signs off, the FFT and control tables for the active files are processed as above in 4.

6. Query references to fields which require label and/or edit mask information are processed by accessing the FFT record for the field to obtain the required information.

When the INCORFFT option is combined with use of the TPQUIPVS load module structure (see section 7.15, TPQUIPVS Load Structure for QUIP), QUIP users will achieve dramatic reductions in setup time when processing signed on files. QUIP setup time is the elapsed time to the start of the file search.

The third operand is valid only for the QUIP component.

SFDSN= TERM / TEMP — Subfile data set name. the two alternate name formats that may be assigned when the QUIP subfile PDS is allocated. The valid keywords are:

TERM - Terminal based name.
name format=terminalid.nnnnnn
example - NMCSSC1.145258

TEMP - OS based name.
name format=SYSnnnnnn.Tnnnnnn.RV000.jobname.SUBFILE
example - SYS76125.T152513.RV000.NIPSMTP.SUBFILE

TERM is the default if no SFDSN parameter is coded.

Note: This option is valid only for QUIP.

Note: At least one QTPSPGM macro is required in the TPSUPGEN job.

Example: The QTPSPGM macro statements for TPQUIP and TPFMSODA, which are included in the standard TPSUPGEN job, have the following formats:

```
QTPSPGM  TPQUIP,(OMQ,INIT,SIGNMLT)

QTPSPGM  TPFMSODA,(OMQ,INIT,SIGN1)
```

TPQUIP uses the output message queue data set, an initializer program named TIQUIP, and is a sign-on program type of which one copy will be loaded for each request. However, since the load module is designated as reenterable only one copy of the load module will be brought into main storage to satisfy the requirements of any number of current tasks. TPFMSODA uses the Cutput Message Queue, requires an Initializer program named TIFMSODA, and is a sign-on program of which only one copy will be loaded and serially reused for all requests.

Other examples of the QTPSPGM macro are:

```
QTPSPGM  TPBLAST

QTPSPGM  TPACCESS,(OMQ,SIGNMLT)

QTPSPGM  TPCOEDIT,(SIGN1)

QPTSPGM  TPEDIT,(OMQ,INIT,SIGN1)

QTPSPGM  TPVIEW,(OMQ,SIGNMLT)

QTPSPGM  TPODE,(OMQ,SIGNMLT)

QTPSPGM   TPQUIPVS,(OMQ,INIT,SIGNMLT,INCORPPT)
```

This example shows the QTPSPGM macro when the TPQUIPVS load structure is required for the QUIP component and control/FFT information is to be maintained in core for signed on files.

b. Specify the TPSUP options using the QTPSOPT macro, with the following format:

```
QTPSOPT [EXTRA={3|nn}] [,SIGNON={NOLOCK|LOCK}] [,DYNAMNT={YES|NO}] [,SUBTASK={YES|NO}]
        [PGROUPS={1|n}]  [,OSEVRN={OS|VS}]      [,ALTIO=name]
```

The operands are keywords, as follows:

EXTRA=3 / EXTRA=nn — where n is the number of empty entries to be added to the program name table. The default value is EXTRA=3. The total number of entries in the program name table cannot exceed 100.

SIGNON=NOLOCK / SIGNON=LOCK — The "lockout" option (SIGNON=LOCK) specifies that requests for major programs will be denied when a Sign-on program is in core. Requests for the minor programs (TPLIST1, TPDUMP) will not be denied. All other programs are considered major programs. This message will be displayed when major program requests are denied -- "XXXXXXXX IS SIGNED ON AND IS THE ONLY MAJOR PROGRAM WHICH MAY CURRENTLY BE RUN." The default value is SIGNON=NOLOCK, in which case

TPSUP will attempt to honor all program requests.

DYNAMNT= YES / NO

The default value is DYNAMNT=YES, in which case the system calls for a mount of user-requested files, on the unit allocated to the DATAFILE DD card. If DYNAMNT=NO is specified, no dynamic mounting (on the unit allocated to DATAFILE DD card in the TPSUP cataloged procedure) will be done.

SUBTASK= YES / NO

The default value is SUBTASK=YES, in which case the system assumes subtasking to be available and uses the ATTACH macro for linking to problem programs. SUBTASK=YES should always be coded for the MVT environment. YES or NO may be coded for the MFT environment without the subtasking capability. SUBTASK=NO should be coded for the MFT environment with subtasking capability if multitasking is not desired within the TP system.

PGROUPS=1 / n

Where n is the number of priority groups desired. The digits 1, 2, and 3 are valid. See the section entitled Time-Slicing for a full discussion of priority groups.

OSEVRN= OS / VS	Identifies the TP operating system environment. The default value is OS. OSEVRN=VS is specified when the operating system environment is VS2, release 2 or later (MVS).
ALTIO=name	Indicates that a user written file access routine with the specified name is to be used by QUIP to read file data records. The routine must be included in the library specified on the STEPLIB DD statement in the TP JCL. This function was developed for a special application and is not designed for general use.

Note: Exactly one QTPSOPT macro statement is required in the TPSUPGEN job, even if all default values are taken. It must be coded after the QTPSPGM macro(s).

Example:

```
QTPSOPT  EXTRA=5,SIGNON=LOCK,DYNAMNT=NO,SUBTASK=NO
```

Five extra entries in the program name table will be generated, the "lockout" option for Sign-on programs will be generated, and no dynamic mounting will be performed.

The following is a sample deck for generating a TP Supervisor similar to the distributed TPSUP.

```
//TPSUPGEN      JOB
//               EXEC    ASMFCL,PARM.ASM=(LOAD,NODECK),PARM.LKED=MAP
//ASM.SYSLIB    DD
              DD       DISP=SHR,DSNAME=FFS.JOBMACRO
//ASM.SYSIN     DD       *
              QTPSPGM  TPQUIP,(OMQ,INIT,SIGNMLT)
              QTPSPGM  TPFMSODA,(OMQ,INIT,SIGN1)
              QTPSPGM  TPBLAST
```

```
            QTPSPGM  TPACCESS,(SIGNMLT)
            QTPSPGM  TPCOEDIT,(SIGN1)
            QTPSPGM  TPEDIT,(OMQ,INIT,SIGN1)
            QTPSOPT  DYNAMNT=NO
            END
/*
//LKED.SYSLIB   DD      DISP=SHR,DSNAME=FFS.JOBLIB
//LKED.SYSLMOD DD DISP=OLD,DSNAME=FFS.JOBLIB(TPSUP)
//LKED.SYSIN DD *
    ENTRY TPSUPEX
    NAME TPSUP(R)
```

The first three cards in the above example need never be modified.

The fourth card overrides the assembly step to concatenate the system data set containing the macros necessary for TPSUP generation.

The QTPSPGM cards in the input stream add TPQUIP, TPFMSODA, TPBLAST,TPACCESS, TPCOEDIT, and TPEDIT to the program name table, and specify the program attributes.

The QTPSOPT macro statement adds three extra entries to the program name table taken, and establishes no dynamic mounting. The lockout option is not generated in the standard TPSUP. Exactly one QTPSOPT macro must be included in the input stream, and it must follow the QTPSPGM macro(s).

The last card in the input stream is the END card which indicates the end of assembly. It must follow the QTPSOPT macro statement.

The last three cards in the TPSUPGEN job apply to the link edit step of the ASMFCL procedure. These cards should appear in the order of the DD statements in that procedure. The DSNAME parameter on the SYSLMOD card points to FFS.JOBLIB which is to contain the generated TPSUP. The DSNAME parameter on the SYSLIB card points to the data set containing the component TP modules comprising the Supervisor. The SYSIN statement indicates that the linkage editor control cards follow in the input stream.

7.6 TP Supervisor Variations

The TPSUP may be modified by the use of PROGLIST and TERMLIST statements at the time the Supervisor is executed. This is done by overriding the SYSIN DD statement with a

```
//TPSUP.SYSIN  DD  *
```

DD statement followed by the PROGLIST and TERMLIST statements.

a. PROGLIST Statements - The PROGLIST card has the following formats:

```
PROGLIST=ALL.
    or
PROGLIST=[USE   ]
         [DELETE] progname progname progname.
```

Any number of program names may be listed, continuing the statement in column 1 of successive cards, ending with a period after the last name.

The PROGLIST statement indicates which of the programs generated into the TPMON name table are to be executed under TPSUP. If all the programs listed on the QTPPROG1, QTPPROG2, and QTPPROG3 cards (see TPMON Generation) are to be marked executable, no PROGLIST card need be supplied since the default is PROGLIST=ALL.

Variations can be made, however, by:

```
PROGLIST=USE progname progname.
              or
PROGLIST=DELETE progname progname.
```

For example:

```
PROGLIST=DELETE  TPFMSODA.
```

will prevent TPSUP from executing a ¬FM request. The program name is the full library name (e.g., TPQUIP,TPMTO1,TPLIST1) including the 'TP' prefix supplied by the Monitor.

b. TERMLIST Statements - The TERMLIST statement has the following formats:

```
TERMLIST=ALL.
     or
TERMLIST= [USE   ]
          [DELETE] termname termname.
```

TERMLIST statements may contain any number of terminal names, continuing the statement in column 1 of successive cards, ending with a period after the last name.

The TERMLIST statement indicates which of the terminals made available at Monitor generation time (see TPMON Generation) are to be serviced by the TP Supervisor.

Terminal names are optionally assigned to hex addresses at Monitor generation (see TPMON Generation). The default name is DDD22X0Z where DDD is the 3-character DD name, X is five or six for 2250 or 2260, respectively, and Z is an A for a 2250, or A, B, C,... for multiple 2260 terminals assigned under one DD card.

If all the available terminals are to be serviced, no TERMLIST card need be supplied since the default is TERMLIST=ALL.

7.7 Multiple TP Supervisors

Multiple TP Supervisors are no longer supported under NIPS Release 21.05.

7.8 TP Monitor/Supervisor Advisory/Diagnostic Messages

The following messages are issued to advise the console operator.

TP TERMINALS OPEN message is written after successful completion of terminal initialization.

TP STANDING REQUEST. REPLIES ARE 'ENA', 'DISA', 'MSG', 'PRTY', 'TPS', or 'TPM'. --Operator can ignore. When it is desired to stop TP Supervisor reply TPS to cause normal end-of-job. When desired to stop the TP application entirely, reply TPM. The reply ENA is used to place a terminal in an active status. The reply DISA is used to place a terminal in an inactive status. The reply MSG is used by the operator to send messages to terminals. The reply PRTY is used to alter terminal priority.

The reply options to the NIPS/TP outstanding request have the following formats:

R nn,	ENA {ALL / name / (name1 name2)}		Enable
	DISA {ALL / name / (name1 name2)}		Disable
	MSG {ALL / name / (name1 name2)}	text of message	Message
	PRTY {ALL / name / (name1 name2)} {1-15 / EXPRESS}		Priority
	TPS		Cancel TP SUPERVISOR
	TPM		Cancel TP MONITOR and SUPERVISOR

TP SUPERVISOR READY -- Message is written after successful completion of TP Supervisor initialization.

The following advisory messages are issued in response to the operators reply to the NIPS/TP outstanding request:

5010 TP MONITOR HAS TERMINATED NORMALLY

The TP Monitor has terminated normally as the result of the computer operator replying TPM to the outstanding TP reply ID. The UTTPDPVR control program will also end as a result of this condition. No user action is required.

5020 TP SUPERVISOR HAS TERMINATED NORMALLY

The TP Supervisor has terminated normally as a result of the computer operator replying TPS or TPM to the outstanding TP reply ID. No user action is required.

5015 TP MON HAS ABENDED. S000 U0000.

The TP Monitor task has ABENDed with a S000 or U0000 error code. The TP Supervisor task will be terminated, and both TPMON and TPSUP will be automatically restarted. If the ABEND persists contact the NIPS Technical Assistance Group for programming support.

5025 TPSUP HAS ABENDED. S000 U0000.

The TP Supervisor task has ABENDed with a S000 or U0000 error code. The TP Monitor will be terminated and both TPMON and TPSUP will be automatically restarted. If the ABEND persists contact NIPS Technical Assistance group for programming support.

3351
5027 TP TERMINAL USERS STILL SIGNED ON. SHOULD TP BE RESTARTED - REPLY YES OR NO.

*** ACTION REQUIRED BY OPERATOR *** The TP Supervisor has terminated as a result of the computer operator replying TPS or TPM to the outstanding TP reply ID; however, terminal users were still active. The operator must reply with one of the following options:

YES The TP Monitor and Supervisor will be restarted.

NO The TP tasks will come to a normal completion, and the UTTPDRVR program will end.

The following diagnostic messages may appear on the operators console any time during NIPS/TP execution:

5027A
5027B NIPS TP LINE ERROR. NO ACTION REQUIRED.

The remote 3270 CRT program (TPBT327R) has detected error conditions. The sense and status message has been acknowledged, and the failing I/O operation re-tried, if appropriate.

5028 TP MONITOR/SUPERVISOR ABEND. MAXWAITS VALUE EXCEEDED.

The TP Monitor/Supervisor task has ABENDed with a S0C1 because the number of multiple wait entries has been exceeded. A high level of system activity caused this ABEND, and the installations's Systems Programmer must increase the number of wait entries to prevent the ABEND from reoccurring.

The following steps must be followed to increase the number of multiple wait entries:

1. Increase the number of multiple wait entries in the QTPEQUS macro. The default value in the distributed system is 100 entries. This value must be increased in the QTPEQUS macro.

2. Assemble and linkedit the following TP programs using the revised QTPEQUS macro.

TPACCESS	TPSUPEX
TPMLIST	TPSUPSTP
TPPAGE1	TPSUPINI
TPSUPCLP	

3. Regenerate the TP Monitor and Supervisor using the assemble programs from step 2.

5029 INVALID PARM FIELD FOR UTTPDRVR

The TP Driver task has abnormally terminated due to an invalid subparameter that has been specified in the PARM field of the EXEC statement. The valid PARM entries are: TPMONxxx, TPSUPxxx, WORKUNIT=, or TP= (see section 6.2 of this document for details). Correct statement and reload TP job.

During the initialization of TP Monitor and TP Supervisor the following advisory diagnostics may appear on the operators console:

TP MONITOR NOT INITIALIZED -- TPSUP task was started before the TPMOM task. Reload the TP job and wait for the message TP TERMINALS OPEN.

2250/2260 WRITE ERROR, IGNORED -- Operator can ignore. Repeated occurrences may indicate a need for IBM CE attention to a problem to in the 2250/2260 hardware.

TPMON SHUTTING DOWN 3277 LINE XX TERMINAL XXXXXX -- TP Monitor task is shutting down specified 3277 terminal due to repeated I/O errors.

TPMON UNABLE TO INITIALIZE 3277 LINE -- Operator can ignore. Diagnostic message is generated when local 3277 assigned to TP job is switched off during TP Monitor initialization.

TOO MANY TERMINALS FOR WAIT LIST -- TERMLIST SYSIN card assigned more than 99 terminals to this TP Supervisor. Reduce the number and reload the TP job. The maximum number is 99.

INVALID KEYWORD IN CONTROL CARD -- SYSIN card does not begin with TERMLIST or PROGLIST. Correct input cards and reload the TP job.

INVALID ITEM REQUESTED - XXXXXXXX -- SYSIN card contained an invalid terminal name or program name, where XXXXXXXX is the name. Correct it and reload TP job.

NO STMT TERMINATOR ON LAST CARD -- SYSIN card has no period after names in list. Insert period, and reload TP job.

BLDL ERROR -- I/O error while searching JOBLIB directory during TP Supervisor initialization. Verify that data set specified on JOBLIB DD statement contains all the named programs and reload TP job.

7.9 TP Monitor/Supervisor ABEND Codes

Code	Explanation
44 45 55	Cause - I/O error on write to a remote 3284/3286 printer device. The printer device may require operator intervention since the I/O error may be caused by an open cover, disable status or being out of paper.
	Action - Check printer, if error persists there may be a hardware problem.
301	Cause - An ECB already has a wait bit set (System ABEND).
	Action - Cancel TP job and reload. If error persists contact NIPS Technical Support Group.
778	Cause - Error in initialization of TP Supervisor.
	Action - Message is on console. Correct and reload TP job.
1234 1235 1237	Cause - I/O/error on read/write to a local 3277 CRT device.
	Action - Check CRT device, if error persists there may be a hardware problem.
2000	Cause - I/O error on read/write of Input Message Queue.

Action - Cancel TP job and reload. If error persists scratch the disk resident IMQ and reallocate.

2004 Cause - Unable to open Input Message Queue.

Action - Cancel TP job and reload.

2008 Cause - I/O error formatting Input Message Queue.

Action - Same as 2004.

2012 Cause - Unsuccessful SPAR (Specify Attention Routine.)

Action - Same as 2004.

2016 Cause - Unable to open statistics data set.

Action - Same as 2004.

3000 Cause - I/O error on write to statistics data set.

Action - Same as 2004.

7.10 TP Monitor/Supervisor and TPSVC Generation Errors

The following MNOTE statements are produced during generation of the TP Monitor, TP Supervisor and TPSVC.

Msg. No.	Severity	Explanation
100	8	QTPSOPT SPECIFIED BEFORE QTPSPGM Action: Place the QTPSOPT macro statement after the QTPSPGM macro, statement and rerun the job.
101	8	MORE THAN ONE QTPSOPT SPECIFIED Action: Do not include more than one QTPSOPT macro. Rerun the job.

INSTALLATION

Msg. No.	Severity	Explanation
102	8	ILLEGAL SIGNON OPTION Action: Change SIGNON= to LOCK or NOLOCK and rerun the job.
103	8	ILLEGAL MOUNTING OPTION Action: Change DYNAMNT= to YES or NO and rerun the job.
104	8	TOTAL NO. OF PGMS EXCEEDS 100 Action: Reduce the number of extra entries for the program name table and rerun the job.
106	8	ILLEGAL ACCOUNTING OPTION Action: Change ACCTNG= to YES or NO and rerun job.
107	8	ILLEGAL FULLPAGE OPTION Action: Change FULLPAGE = to YES or NO and rerun job.
108	8	VALIDATION ROUTINE NAME TOO LONG Cause: The name of the specified installation validation subroutine exceeds 8 characters in length. Action: Correct the name and rerun the job.
110	8	ATTRIBUTE IS ILLEGAL PARAMETER Action: Verify that all program attributes are spelled correctly and rerun the job.

INSTALLATION

Msg. No.	Severity	Explanation
111	8	ATTRIBUTE IS INCONSISTENT PARAMETER Cause: The specified attribute and at least one other attribute are mutually exclusive. Action: Correct the program attributes and rerun the job.
112	8	NO PARAMETERS SPECIFIED Cause: No operands are coded on the QTPSPGM macro. Action: Specify the desired program and attributes and rerun the job.
113	8	ONLY ONE PROGRAM REQUEST PER QTPSPGM MACRO Action: Specify the desired programs on separate QTPSPGM macro statements and rerun the job.
114	Ø	INCLUSION OF PGM IS AUTOMATIC, REQUEST IGNORED Cause: You specified TPLIST1, TPMT01, TPDUMP, or GGINTFAC on a QTPSPGM macro statement. This macro statement has been ignored. Action: None.
115	12	ILLEGAL NUMBER OF IMQ RECORDS SPECIFIED. MUST BE BT 10/99. Cause: An incorrect value was specified for the IMQ operand of the QTPMOPT macro. The

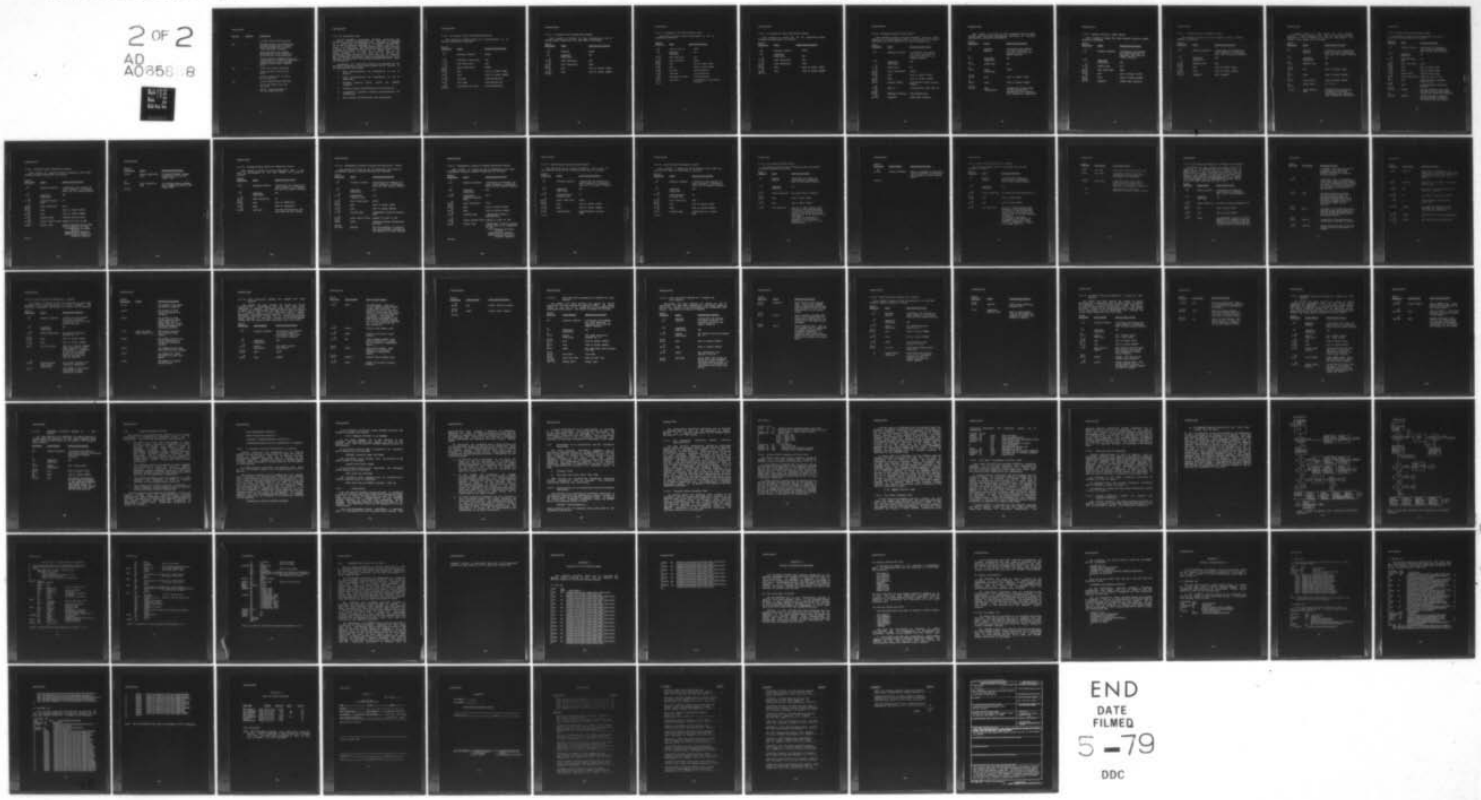
Msg. No.	Severity	Explanation
		value was not DASD nor was it a number between 10 and 99. Action: Specify DASD if the IMQ is a data set allocated on a direct access storage device. Specify a 2 digit number between 10 and 99 (inclusive) to indicate the number of incore IMQ lines to be allocated internally for each terminal in the TP system.
120	8	PRIORITY EXCEEDS RANGE Cause: The PRTY= value specified was not within the range of 1 to 15. Action: Change PRTY= to a valid priority and rerun job.
121	0	PRIORITY OMITTED, DEFAULT USED Cause: PRTY= was not specified. Action: None if priority 1 is desired.
122	8	PGROUPS VALUE EXCEEDS RANGE Cause: The PGROUPS= value specified was not within the range of 0 to 3. Action: Change PGROUPS= to a valid value and rerun job.
123	0	PGROUPS OPERAND OMITTED DEFAULT TO ZERO SUBSTITUTED Cause: PGROUPS= was not specified.

Msg. No.	Severity	Explanation
		Action: None if priority groups are not desired.
124	8	PRIORITY 'sublist entry' EXCEEDS RANGE Cause: The PRTY= sublist entry priority value specified was not within the range of 1 to 15. Action: Change the entry to a valid priority value.
125	Ø	PRTY 'sublist entry' OMITTED, DEFAULT USED Cause: The PRTY sublist entry was not specified. Action: None if priority 1 is desired.
126	0	THE REQUIRED QTPSPGM MACRO OPTION(S) 'option', WERE SET FOR COMPONENT 'component name'. Cause: the required TP Supervisor program 'option(s)' were set for the NIPS component specified as 'component name'. Action: None.
200	Ø	2250 NEEDS 1 DD PER UNIT Action: Change UNITS= to 1 and rerun the job.
201	Ø	ILLEGAL TYPE Cause: Unsupported terminal type. Action: Change TYPE= to the correct

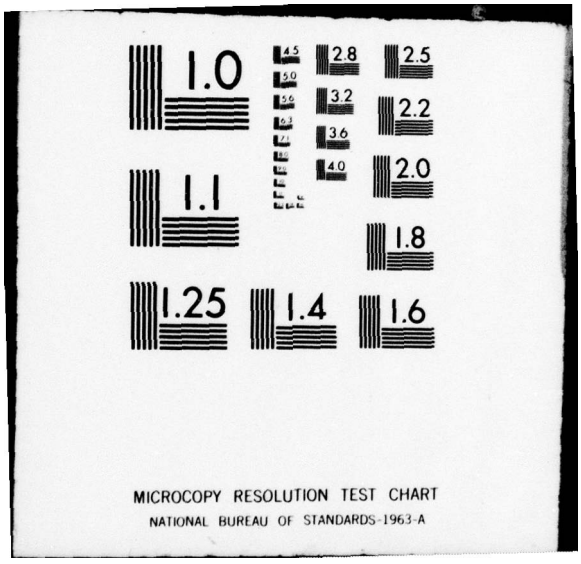
AD-A065 888 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
SEP 78

UNCLASSIFIED CCTC-TR-54-78 SBIE-E100 124 NL

2 OF 2
AD
A065888



END
DATE
FILMED
5-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Msg. No.	Severity	Explanation
		terminal type and rerun the job.
202	8	The name field of the QTPDD and/or. QTPLINE macro(s) which is used to define the DDNAME may not exceed three characters in length. Cause: The name field (DDANME specification) on the QTPDD macro was longer than three characters. Action: Specify a DDNAME of from 1 to 3 characters in the QTPDD and associated QTPLINE macro(s). Note: only BTAM terminal types require a QTPLINE macro.
220	8	INVALID CP67 OPTION SPECIFIED Cause: The CP67= entry was not valid. Action: Change CP67= to YES or NO and rerun job.
221	8	INVALID GAM2250 OPTION SPECIFIED Cause: The GAM2250 entry was not valid. Action: Change GAM2250= to YES or NO and rerun job.

7.11 TP Accounting Data

The TP component optionally provides accounting and statistics data for installation use. This data is recorded in a single data set (defined by the STATRECS DD statement in the TP Job Control Language) in a format suitable for input to a NIPS file. The NIPS.SAMPLE.JOB library contains a sample job (TPLOGJOB) which could be used to create a NIPS file from this data. All data is in EBCDIC and can be output directly to a print data set if desired. The accounting output directly to a print data set if desired. The accountand statistical data is accumulated from the TP Monitor, TP Supervisor and problem programs executing under the Supervisor.

Accounting and statistics records are recorded for the following functions as they occur within the TP Monitor or TP Supervisor(s) partitions:

a. Task initialization and termination of the TP Monitor

b. Task initialization and termination of the TP Supervisor

c. Terminal operator LOGON, LOGOFF and REMARKS requests

d. Problem program initialization and termination.

e. Independent graphics program initialization and termination.

f. PAGE program initialization and termination.

7.11.1 TP Monitor Task Initialization Record

This record is output during the initialization of TP Monitor partition as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal address	Blank
6		
7-9	Component Identifier	TPM
10		
11-13	Task Identifier	JOB
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format *YYDDD*
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Job Name	Self-explanatory
43		
44-51	Step Name	Self-explanatory
52		
53-60	Procedure Step Name	Self-explanatory
61-133		

7.11.2 TP Monitor Task Termination Record

This record is output at the termination of the TP system (after Supervisor has completed) as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal	Blank
6		
7-9	Component Identifier	TPM
10		
11-13	Task Identifier	JOB
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-133		

7.11.3 TP Supervisor Task Initialization Record

This record is output during initialization of the TP Supervisor as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	Blank
6		
7-9	Component Identifier	TPS
10		
11-13	Task Identifier	JOB
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Job Name	Self-explanatory
43		
44-51	Step Name	Self-explanatory
52		
53-60	Procedure Step Name	Self-explanatory
61		
62-64	Task DPRTY	TP Supervisor's dispatching priority.
65-133		

7.11.4 TP Supervisor Task Termination Record

This record is output by the TP Supervisor during termination processing as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	Blank
6		
7-9	Component Identifier	TPS
10		
11-13	Task Identifier	JOB
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-133		

7.11.5 Terminal Operator LOGON Record

This record is output for each terminal operator LOGON request in a TP environment where no installation validation subroutine has been specified at monitor generation.

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPM
10		
11-13	LOGON Code	LOG
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-35		
36-45	Account Number	10-character LOGON Account Number
46		
47-56	User ID	10-character LOGON user ID
57		
58-61		
62-64	Terminal Priority	Self-explanatory
65		
66-133	Comments	LOGON user comments

INSTALLATION

This record is output for each terminal operator LOGON request which has been successfully validated by optional installation validation subroutine.

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal, 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPM
10		
11-13	LOGON Code	LOG
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-46		
47-115	User Information	Maximum of 70 characters of LOG information provided by the installation validation subroutine.

7.11.6 Terminal Operator LOGOFF Record

This record is output for each terminal operator LOGOFF request as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPM
10		
11-13	LOGOFF Code	LOG
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-57		
58-133	Comments	LOGOFF user comments

7.11.7 Terminal Operator REMARKS Record

This record is output for each terminal operator REMARKS request as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPM
10		
11-13	REMARKS Code	REM
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34-57		
58-133	Comments	User comments

A REMARKS record is also output for each terminal operator. LOGON request that has not been successfully validated by the installation validation subroutine. The format of the record is as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPM
10		
11-13	REMARKS Code	REM
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-55	LOGON Error	LOGON VALIDATE ERROR-
56	Error Level	1, 2, or 3
57-58		
59-128	User Information	Maximum of 70 characters of LOGON information provided by the installation validation subroutine.

7.11.8 Problem Program Initialization Record

This record is output by the TP Supervisor just prior to the Problem program execution as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Problem Program Code	PP
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Program Name	Problem program name
43		
44-48	Input Record Count	Number of lines on IMQ
49		
50-55	Core	Length of largest available block of core
56-61		
62-64	DPRTY	Terminal/program dispatching priority
65-118		
119-125	LOCKOUT	The word LOCKOUT if an attempt was made to load a second major program when LOCKOUT was specified
126		
127-133	EXPRESS	The word EXPRESS if terminal was given express priority by the operator for this request.

7.11.9 Problem Program Termination Record

This record is output by the TP Supervisor just after problem program termination as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Problem Program Code	PP
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Program Name	Problem program name
43		
44-48	Output Record Count	Number of lines on OMQ
49		
50-54	Return Code	Problem program return code if it completes; if not: 11111=program was never started ABEND=program ABEND'ed ABORT=program aborted by terminal operator
55-56		

INSTALLATION

Record Positions	Field	Description/Content
57-60	System Completion Code	If system ABENDed system completion code in the format SXXX
61		
62-66	User Completion Code	If problem program ABENDed, user completion code in the format UXXX
67-133		

7.11.10 Problem Program Batch Job Submitted Record

This record is output by the TP Supervisor when a JOB card is encountered in the EDIT SUBMIT job stream as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Task Identifier	BJ
14-21		
22-26	Date	Date of submission
27		
28-33	Time	Time of submission
34		
35-114	Job Card	The full 80-character JOB card from the job stream.

7.11.11 Independent Graphics Program Initialization Record

This record is output by the TP Supervisor just prior to Independent Graphics program execution as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Independent Graphics Code	IGP
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Program Name	Independent Graphics program name
43		
44-48	Input Record Count	Number of lines on IMQ
49-61		
62-64	DPRTY	Terminal/program dispatching priority
65-126		
127-133	EXPRESS	The word EXPRESS if terminal was given express priority by the operator for this request.

7.11.12 Independent Graphics Program Termination Record

This record is output by the TP Supervisor just after Independent Graphics program termination as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Independent Graphics Program Code	IGS
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Program Name	Independent Graphics program name
43		
44-48	Output Record Count	Number of lines on OMQ
49		
50-54	Return Code	Independent Graphics program return code if it completes; if not: 11111=program was never started ABEND=program ABEND'ed ABORT=program aborted by terminal operator
55-133		

7.11.13 PAGE Program Initialization Record

This record is output by the TP Monitor just prior to the terminal entering the conversational mode as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Conversational Program Code	CP
14		
15-19	Start Identifier	START
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	program Name	Conversational Program name
43-133		

7.11.14 PAGE Program Termination Record

This record is output by the TP Monitor just after the conversation mode is terminated as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal
6		
7-9	Component Identifier	TPS
10		
11-13	Conversational Program Code	CP
14		
15-19	Stop Identifier	STOP
20-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-42	Program Name	Conversational program name
43-133		

7.11.15 QUIP Source Statement Record

This record is output by the TP Supervisor when QUIP reads a user source statement as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for Local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	QSS (QUIP Source Statement)
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	Run Identifier	This is a 7-digit numeric field assigned to each QUIP execution. It consists of the two low order digits of the Julian day and the starting time of the QUIP execution in seconds. This same run identifier is assigned to QUIP Statistics Messages No. 1 and No. 2 of the same execution.

INSTALLATION

Record Positions	Field/Header	Description/Content
42		
43-122	Source Statement	This is a display of the user's source statement as entered at the terminal or from a stored query.
123-133		

7.11.16 QUIP Statistics Message No. 1 Record

This record is output by the TP Supervisor for each QUIP execution as follows:

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	QS1 (QUIP Statistics Message No. 1)
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	Run Identifier	This is a 7-digit numeric field assigned to each QUIP execution. It consists of the two low order digits of the Julian day and the starting time of the QUIP execution in seconds. This same run identifier is assigned to the QUIP Source Statement Records and to QUIP Statistics Message No. 2 Records of the same execution.

INSTALLATION

Record Positions	Field/Header	Description/Content
42		
43-86	File Name	44-character field for the file name
87-118	File Class	32-character field for the file classification unless suppressed by the operand 'N' in the query
119		
120-123	Query Number	4-character field for the query number (space is provided for this field but a query number is not applicable for online QUIP executions).
124		Query Number and RITID field separator and locator (always printed).
133		

7.11.17 QUIP Statistics Message No. 2 Header and Data Records

The header and data records are output by the TP Supervisor as a consecutive pair for each QUIP query. The header record is put out only as a visual aid to interpreting the data record when the output is a printed copy. In the following record format, positions 1-34 describe both records; in positions 35-133, the Record Positions column and the Description/Content column pertain to the data record, and the Field/Header column will contain the actual header associated with that field.

Record Positions	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	QS2 (QUIP Statistics Message No. 2)
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	RUNID	Run Identifier - This is a 7-digit numeric field assigned to each QUIP execution. It consists of the two low order digits of the Julian day starting time of the QUIP execution

Record Positions	Field/Header	Description/Content
		in seconds. This same run identifier is assigned to the QUIP Source Statement Records and the QUIP Statistics Message No. 1 Record of the same execution.
42 43-44	RC	Return Code - This is a 2-character field derived from a bit ORed composite of QUIP error message severity levels. For example, if RC-09, it would indicate that there was at least one level-8 and one level-1 error message for the QUIP execution.
45 46-50	LINES	This entry specifies the number of lines written by QUIP to the OMQ (Output Message Queue). If OMQ=NO was specified, the value is the number of lines written to the VIEW data set member, or zero if a VIEW data set member was not specified.
51-52 53-55	WAITS	The number of 1.0 second wait periods executed by QUIP when attempting to get main storage that was unavailable. Each wait period is preceded by an unsuccessful GETMAIN.
56-57 58-66	QUIP-ET	Elapsed time of the QUIP execution, in seconds and hundredths of seconds.
67 68-76	SETUP-ET	Elapsed time to the start of the file search in seconds and hundredths of seconds.

Record Positions	Field/Header	Description/Content
77		
78-86	OUTST-ET	Output Start Elapsed Time. The elapsed time in seconds and hundredths of seconds from QUIP start time to the availability of the first line of output.
87		
88-95	CPU-TIME	Total CPU time included in the total elapsed time.
96		
97-102	RPL	The number of records passing the limits or retrieved.
103		
104-109	RQ	The number of qualifying records.
110		
111	L	Limit statement used indicator. Y=yes, N=no.
112		
112-118	CANDTS	The number of candidates if the retrieval were indexed; if not, the field will contain 'NA'.
119		
120-125	R-EXCP	Retrieval EXCP count (I/O operations)
126		
127-132	S-EXCP	Sort EXCP count (I/O operations)
133		

7.11.18 QUIP Statistics Message No. 3 Record

This record is output by the TP Supervisor for each QUIP execution with a query containing a SORT statement. The following describes the format and content of the message.

Record Positions	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	QS3 (QUIP Statistics Message No. 3)
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	Run Identifier	This is a 7-digit numeric field assigned to each QUIP query. It consists of the two low order digits of the Julian day and the starting time in seconds.
42		
43-74	Process Block Disk Space Statistics	This field contains the following subfields:
56-59		The number of PB tracks specified on first allocation attempt.

Record Positions	Field	Description/Content
61-63		The number of PB space allocation attempts.
65-68		The number of disk tracks allocated for PB space.
70-73		The number of disk tracks actually used. This number may be larger than the number allocated because of secondary extents.
76-77	Sort Key Disk Space statistics	This field contains the following sub-fields.
80-83		The number of sort key tracks specified on the first allocation attempt.
85-87		The number of sort key space allocation attempts.
89-92		The number of tracks allocated for sort key space.
94-97		The number of tracks actually used.

7.11.19 QUIP Statistics Message No.4 Header and Data Records

The header and data records are output by the TP Supervisor as a consecutive pair for each query which processes a subfile. The header record is put out only as a visual aid to interpreting the data record when the output is a printed copy. In the following record format, positions 1-34 describe both records. In positions 35-133, the Record Positions column and the Description/Content column pertain to the data record, and the Field/Header column show the actual header associated with that field.

Record Positions	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	QS4 (QUIP Subfile Statistics)
14-21		
22-26	Date	YYDDD
27		
28-33	Time	HHMMSS

Record Positions	Field/Header	Description/Content
34		
35-41	RUNID	Run Identifier - This is a 7-digit numeric field assigned to each QUIP execution. It consists of the two low order digits of the Julian day starting time of the QUIP execution in seconds. This same run identifier is assigned to the QUIP Source Statement Records and the QUIP Statistics Message No. 1 and 2 Records of the same execution.
42		
43-49	INPUT---	Subfile input member name.
50		
51-55	COUNT	Number of entries in input subfile.
56		
57-59	SEQ	Input sequence number. Each subfile created in a unique PDS is assigned a consecutive number.
60-61		
62-64	LEVEL	Input level number. Each subfile in one chain is assigned a consecutive number.
65-69		
70-76	OUTPUT---	Subfile output member name.
77		
78-82	COUNT	Number of entries in output subfile.

INSTALLATION

Record Positions	Field/Header	Description/Content
83		
84-86	SEQ	Output sequence number.
87-88		
89-91	LEVEL	Output level number.
92-133		

7.11.20 SODA Statistics Message No. 0 Header and Data Records

The header and data records are output by the TP Supervisor for each successful signon to SODA. The header record is put out as a visual aid for interpreting the subsequent data record. The record is output as follows:

Record Position	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	FM0 (SODA Statistics Message No. 0)
14-21		
22-26	Date	Date in format YYMMDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	RUNID	Run Identifier (see Section 7.11.19)
42-43		
44-87	FILE NAME	File Name
88-89		
90-101	HOLD FILE NAME	Name of hold file
102-106		
107-114	REPORT NAME	Report name

7.11.21 SODA Statistics Message No. 1 Header and Data Records

The header and data records are output by the TP Supervisor for each Soda transaction. The header record is put out as a visual aid for interpreting the subsequent data record. The record is output as follows:

Record Position	Field	Description/Content
1	Terminal Address	3-character UCB address for local terminal; 4-character POLL-list for remote terminal.
2-5		
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	FM1 (SODA Statistics Message No. 1)
14-21		
22-26	Date	Date in format YYMMDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	RUNID	Run Identifier (see Section 7.11.19)
42-46	ERR-CODE	Error Code. This value is 00 if the logic statement executed without error. If the input was retained for correction this value will be 04.
47-48		

Record Position	Field	Description/Content
49-57 58-62	BLDPB-ET	Build Process Block Elapsed Time. This is the elapsed time in hundredths of seconds from the start of SODA to the availability of the process block.
63-67 68-72	LS-ET	Logic Statement Elapsed Time. This is the elapsed time in hundredths of seconds for the execution of the user's logic statement.
73-77 78-81	UP-ET	Update Elapsed Time. This is the elapsed time in hundredths of seconds for post logic statement processing. Currently, this value will appear as 000 since all update records are placed on the hold file.

7.11.22 SODA Statistics Message No. 2 Record

This record is output by the TP Supervisor for each SODA UPDATE or CANCEL command as follows:

Record Position	Field	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL-list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	FM2 (SODA Statistics Record No. 2)
14-21		
22-26	Date	Date in format YYMMDD
27		
28-33	Time	Time in format HHMMSS
34		
35-41	RUNID	Run Identifier (see Section 7.11.19)
42-45		
46-54	U/C IND	Field Label-Update/Cancel Indicator
55	Update/Cancel Indicator	This field will contain a 'U' or 'C' indicating whether the user has entered an update or CANCEL command

Record Position	Field	Description/Content
56-60		
61-69	EXEC-ET	Field Label-Execution Elapsed Time
73-78	Execution Elapsed Time	This is the elapsed time in hundredths of seconds to perform the UPDATE or CANCEL.

7.11.23 FORMATTER Statistics Message No. 1 Header and Data Records

The header and data records are output by the TP Supervisor as a consecutive pair for each format processed by FORMATTER. In the following record format, positions 1-34 describe both records; in positions 35-110, the Record Positions column and the Description/Content column pertain to the data record, and the Field/Header will contain the actual header associated with that field.

Record Positions	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	OS1 - header record OS2 - data record
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-74	User Library DSNAME	User Library Data set name. The name of the data set that contains the format being accessed.
75-79		
80-87	FORMAT	Format. The name of the format being accessed.
88		
89-94	ACCESS	Access Elapsed Time. This is the elapsed time in hundredths of seconds needed to load the format.

Record Positions	Field/Header	Description/Content
95		
96-101	EXEC	Execution Elapsed Time. This is the elapsed time in hundredths of seconds for the processing of the formats.
102-103		
104-105	CHAIN	Number of Chained Formats. This field contains the number of formats accessed prior to executing the invoked component.
106-108		
109-110	ERR	Number of Error Retries. This field contains the number of retries necessary to complete all the data required fields.

7.11.24 FORMATTER Statistics Message No. 2 Header and Data Records

The header and data records are output by the TP Supervisor as a consecutive pair for each format UPDATE processed by FORMATTER. In the following record format, positions 1-34 describe both records; in positions 35-132, the Record Positions column and the Description/Content column pertain to the data record, and the Field/Header will contain the actual header associated with that field.

Record Positions	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identifier	OS3 - header record OS4 - data record
21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-68	Source Input Library	Source Input Library Data set name. This field contains the name of the source input library.
69		
70-77	Input Member	Input Member Name. This field contains the name of the input source member.
78		
79-112	Format Output Library	Format Output Library Data Set name. This field contains the name of the output library, where the format was stored.

Record Positions	Field/Header	Description/Content
113		
114-121	Output Member	Output Member Name. This field contains the name of the output member.
122		
123-126	Size	Size of Format. This field contains the size in bytes of the format.
127		
128-132	EXEC	Execution Elapsed Time. This is the elapsed time in hundredths of seconds needed for the definition and storage of the format.

7.11.25 FORMATTER Statistics Message No. 3 Data Records

If, when signing on to FORMATTER, the debug snapshot is invoked (0256 ¬0), each line of the Input Message Queue (IMQ) created by each format will be logged in the following record format:

Positions	Field/Header	Description/Content
1		
2-5	Terminal Address	3-character UCB address for local terminal; 4-character POLL list for remote terminal.
6		
7-9	Component Identifier	TPS
10		
11-13	Record Identified	OS5 - data record
14-21		
22-26	Date	Date in format YYDDD
27		
28-33	Time	Time in format HHMMSS
34		
35-114	IMQ	One line of IMQ created by the format identified in the preceding FORMATTER Statistical Record #1. This record format will be repeated until the entire IMQ has been written.

7.12 Terminal/Program Priority

The concept of terminal/program affords the user maximum flexibility in establishing priorities within the TP environment. Four priority environments are available:

a. The "first in, first out" environment in which problem program requests are processed on a first in, first out basis, i.e., when a request is entered, it assumes higher priority than all subsequent requests. This environment is established by assigning equal terminal priorities and equal problem program priorities when generating the TP Monitor and TP Supervisor, respectively. Allowing all PRTY specifications to assume their default values will result in this environment.

b. The terminal priority environment in which requests are processed in order of relative terminal priority. This environment is established by assigning relative terminal priorities and equal problem program priorities when generating the TP Monitor and TP Supervisor, respectively.

c. The problem program priority environment in which requests are processed in order of relative priority of the problem program requested.

d. The terminal and program priority environment in which requests are processed in order of relative terminal/program priority combinations.

The four priority environments are achieved within one priority scheme, that of terminal/program priority combinations. The TP Supervisor uses the terminal/program priority combination to compute the dispatching priority of the request. Requests are queued and dispatched in dispatching priority order. Dispatching priorities are derived as follows:

Task dispatching priority =

TPSUP dispatching priority -

((highest terminal/program priority +1) -

(terminal/program priority combination of request))

7.12.1 TP Terminal Priority Modification by Operator

Terminal priorities are established at TP Monitor generation time and are the priorities in effect when the TP Monitor is initialized. Once the TP Monitor has been initialized, terminal priorities may be changed by the computer operator. Terminal priorities so assigned remain in effect until subsequently changed again or the TP Monitor is reinitialized.

To change terminal priorities, the operator must reply to the TP standing request that is always issued by the TP Monitor.

The operator responds to this request with the keyword PRTY followed by the terminal name, terminal names enclosed within parentheses, or ALL and the new priority value. ALL will cause all terminal priorities to be changed to the new priority value. Terminal names or ALL must be separated from the priority value by one or more blanks or commas. All characters within the reply must be uppercase. The priority value must be numeric and has a range of 1 to 15 or the word 'EXPRESS'. EXPRESS may not be specified with ALL. Express priority will give the terminal(s) the highest priority for one TP problem program request. A successful priority change will be indicated by the following response to the operator:

TERMINAL(S) PRIORITY CHANGE COMPLETED

At the terminal receiving a newly assigned priority, the following message will appear:

***** TERMINAL PRIORITY IS XX EXPRESS

In the above example, XX is the priority of the terminal. The word EXPRESS will appear as part of the message only when the operator assigns express priority to the terminal.

If an invalid terminal name is specified, the following error message will be returned:

TERMINAL (terminal name) NOT FOUND

The following error message will be returned if the reply format is incorrect:

INVALID PRTY REPLY FORMAT

If the priority value is not specified, the following error message will be returned:

PRTY VALUE NOT SPECIFIED

The following error message will be returned if an invalid priority value is specified:

PRTY VALUE MUST BE NUMERIC (VALUES 1 THRU 15)

7.13 Time-Slicing

The time-slicing facility allows the user to establish a group of tasks or partitions (called the time-slice group) that are to share the use of the CPU; each for the same, fixed interval of time. Implementation of this facility differs slightly in MFT, MFT with subtasking, and MVT systems. The resultant capability operates the same in all systems. The difference occurs in the generation of OS for time-slicing.

In an MFT environment without subtasking, a partition must be established for each priority group desired. This

partition must also contain a resident TP Supervisor. Execution of tasks in these partitions is based upon the inclusion or exclusion of specific programs and terminals at TP generation time. Priority in this case may be recognized only by partitions and overrides the group concept defined above.

To facilitate the interface with OS Time-Slicing, the task dispatching priorities derived by the method set forth in section 7.8, Terminal/Program Priority, will be ordered into several groups. Tasks within each group will be attached at the same OS job priority. Group membership for a particular terminal/program priority is determined in the following manner:

a. If only one group is specified by the user, all tasks in the TP environment are attached at a dispatching priority 16 less than that of TPSUP. This is equivalent to decrementing the job priority of the task by one. That is, if TPSUP is a priority 13 job, all tasks will be considered the same as priority 12 jobs.

b. If two groups are requested, the highest terminal/program priority in the system is halved and the result is truncated. All terminal/program priority combinations greater than this limit are members of Group 1 and are attached with a dispatching priority 16 less than that of TPSUP. All other combinations are placed in the second group and attached with a dispatching priority 32 less than TPSUP (i.e., two OS job priorities less). Either, both, or neither of these groups may be time-sliced at the user's option.

c. If three groups are requested, the limit derived in Step 2 is halved again, the result becoming the lower bound for Group 2. All combinations equal to or less than this new limit are placed in Group 3 and attached with a dispatching priority 48 less than that of TPSUP (3 OS job priorities less). Any combination of groups may be time-sliced or not time-sliced at the user's direction.

A slight modification to the above method is necessary in an MFT environment (with or without subtasking). The groups established will be attached at 11, 22, or 33 dispatching priorities less than TPSUP. This is equivalent to a difference of 1, 2, or 3 partition priorities. The remainder of the above discussion is pertinent in all cases.

7.14 Development of an Installation Written Validation Subroutine

The NIPS Terminal Processing Component and TP application programs contain the linkage needed to interface with an installation validation subroutine, written by the user, designed to control the rightful access to the TP terminal devices and data files. The user written validation subroutine is optional and may be as simple or complex as the installation requires. If the optional validation subroutine is specified by the user, an exit to such a subroutine will be made at the following points in time:

1. Terminal LOGON

2. File Open (for QUIP, SODA, ODE, VIEW)

When writing the installation validation subroutine certain conventions must be followed. The following subsections describe such conventions.

7.14.1 Specification of an Installation Written Validation Subroutine

The presence of an installation validation subroutine is indicated by specifying the appropriate keyword option on the QTPMOPT macro at TP Monitor generation time. To specify an installation validation subroutine using the QTPMOPT macro, the following format is used:

```
QTPMOPT  VALID=xxxxxxxx,...
```

where xxxxxxxx is the 1-8 character entry point name of the validation subroutine.

The installation validation subroutine will be included at linkedit time as a part of the generated TP Monitor module. For a complete description of the QTPMOPT macro, see section 7.4 of this manual.

7.14.2 The Validation Subroutine Program Interface Specifications

The user written installation validation subroutine should follow the standard OS/360 linkage and NIPS Subloader conventions as specified in section 3 of Volume I, Introduction to File Concepts. Three parameters are passed to the user routine. Parameter one is the entry point to the NIPS subroutine loader. Parameter two points to the parameter area which is described in detail in the following sections. Parameter three is a cell for return code storage. The subroutine loader entry point is provided to the user validation subroutine so that requests to load or link to other routines and/or tables are possible. The high order byte of parameter two contains either L or F indicating whether the subroutine is being branched to at terminal LOGON or File Open time, and thus determines whether the parameter area has the format of LOGON or File Open data. The one character cell designated by parameter three, or register 15, can contain a return code of 0, 1, 2, or 3. The installation written validation subroutine should be compiled and linkedited to FFS.JOBLIB, so it is available at TP Monitor generation time. The installation validation subroutine should not be loaded to FFS.JOBLIB by the NIPS SUBLOADER procedure.

7.14.3 Terminal LOGON: Parameter Area

At terminal LOGON the parameter area pointed to by parameter two will contain the LOGON data, exclusive of the keyword LOGON, a cell for the encoded data value resulting from a successful LOGON, the terminal device type, the terminal's UCB address, the time of LOGON attempt and areas for optional messages to the terminal user and console operator. By using the macro QTPLGDEF and coding a USING TPLGDEF instruction in the validation subroutine, the following fields can be referenced:

```
TPLGDEF  DSECT
LGDATA   DS    CL70    TERMINAL INPUT EXCLUDING KEYWORD LOGON START
*                      DATA WRITTEN ON TPLOG DATASET AS REM OR LOG START
LGCODE   DS    CL4     ENCODED VALUE - PROPER LOGON OCCURRED
LGDEVTYP DS    CL1     DEVICE TYPE
*                    X'01' = LOCAL 2260
*                    X'02' = LOCAL 2250
*                    X'10' = REMOTE 2260
*                    X'20' = DIAL=UP 1050
*                    X'40' = 2741
*                    X'80' = 3270
LGDEVNUM DS    CL3         UCB ADDRESS
LGTIME   DS    CL4         LOGON TIME HHMM
LGMGTERM DS    CL70        OPTIONAL MSG TO TERMINAL OPERATOR
LGMGCON  DS    CL70        OPTIONAL MSG TO CONSOLE OPERATOR
```

7.14.4 Terminal LOGON: Consequence of Return Codes

Upon return from the user written validation subroutine register 15 or the one character cell designated by parameter three, can contain a hex value of 0, 1, 2, or 3.

A return code of 0 indicates that the terminal user has supplied valid accounting data and the terminal is to logged on. The data in the LGDATA area of the TPLGDEF DSECT is written out on the TPLOG data set in a LOGON START record, positions 47-115. The user subroutine may overlay the actual LOGON data with an unclassified version of the information. The encoded value placed in the LGCODE area is stored in the Terminal's Unit Status Table so that it will be available at file open.

A return code of 1 indicates that the terminal user has supplied incorrect accounting data and that the terminal is not to be logged on. The error, however, is considered to be minor and the user is allowed to attempt to LOGON again. The data in the LGDATA area is written out on the TPLOG data set in a REMARKS record, positions 58-127. The standard message LOGON INFORMATION MISSING OR INCORRECTLY SPECIFIED is written out to the terminal user and followed with the optional message, is specified in the LGMGTERM area.

A return code of 2 indicates that the terminal user has supplied incorrect accounting data and that the terminal is not to be logged on. The error is considered to be serious, but the user is allowed to LOGon again. A count of serious errors incurred by each terminal is maintained. The third consecutive occurrence of a return code of 2 will cause any further input from that terminal to be disregarded until (the system manager instructs) the computer console operator to enable the terminal. The data in the LGDATA area is written out on the TPLOG data set in a REMARKS record, positions 58-127. The standard message LOGON INFORMATION MISSING OR INCORRECTLY SPECIFIED is written out to the terminal user and followed with the optional message, if specified in the LGMGTERM area.

A return code of 3 indicates that the terminal user has supplied incorrect accounting data and that the error is considered to be gross in nature. The terminal is immediately placed in a disabled status. The terminal user is sent the standard message, TERMINAL DISABLED UNTIL FURTHER NOTICE, and an optional message if specified in the LGMGTERM area. The computer console operator is informed that UNAUTHORIZED USER ATTEMPTING TO USE TERMINAL xxxxxxxx. If an optional message is given the LGMGCON area it is written to the console operator after the standard message. The data in the LGDATA area is written out on the TPLOG data set in a REMARKS record positions 58-127. Any further attempt to enter data by the terminal operator will result in the issuing of the following message: NO INPUT ALLOWED - TERMINAL IS IN DISABLED STATUS. The computer console operator can enable the terminal again by replying to the NIPS TP outstanding request:

R nn, 'ENABLE terminal name'

7.14.5 File Open: Parameter Area

At File Open the parameter area will contain the file name, the member name (optional), the file classification, the encoded value resulting from successful LOGON, a cell for the file access code, and areas for optional mesages to the terminal user and console operator. By using the macro QTPFLDEF and coding a USING TPFLDEF instruction in the

validation subroutine the following fields can be referenced:

```
TPFLDEF  DSECT
FLNAME   DS        CL44      DATA SET NAME
FLCLASS  DS        CL32      FILE CLASSIFICATION
FLLGCODE DS        CL4       ENCODED VALUE OF SUCCESSFUL LOGON
FLACES   DS        CL4       FILE ACCESS CODE
*                  CL1       R/B/W READ,WRITE, OR BOTH
*                  CL1       A/M ALL/MEMBER
*                  CL2       RESERVED FOR FUTURE USE
FLPGM    DS        CL8       READ/WRITE ACCESS
FLMGTERM DS        CL70      OPTIONAL MSG TO TERMINAL OPERATOR
FLMGCON  DS        CL70      OPTIONAL MSG TO CONSOLE OPERATOR
FLMEM    DS        CL8       MEMBER NAME
```

7.14.6 File Open: Consequence of Return Codes

Upon return from the user written validation subroutine register 15 of the one character cell designated by parameter three can contain a hex 0,1,2, or 3. Return codes of 1 or 2 will be treated as 0 until read only, write only, and limited access capabilities are implemented.

A return code of 0 indicates that the terminal user has rightful access. The first byte returned in FLACES will be either R (read), W (write), or B (both). This value is stored in the MCT and can be checked at a later time to control read/write access to the NIPS file. The second byte returned in FLACES will be either A (all) or M (member) if the data set being validated is a library. Validation of members is done on an exception bases: if A is specified all members in the library have the same access code and no additional validation is performed when specified members are referenced. In QUIP, no further checking of file access is performed, in SODA, the R/W/B indicator is checked at file update time, in VIEW and FORMATTER, the A/M indicator is checked at library open time to determine whether further validation is required on a member name bases.

A return code of 3 indicates that the terminal user does not have rightful access to the data file and the TP application program is terminated immediately. The

standard message APPLICATION PROGRAM TERMINATED DUE TO INVALID ACCESS is written out immediately to the terminal user, and followed by an optional message if specified in the FLMGTERM area. In addition the optional message defined by FLMGTERM may be used by the TP application program to format error messages. The standard message of TERMINAL xxxxxxxx UNAUTHORIZED TO ACCESS FILE xxxxxxxxxx (maximum of 44 characters) is written out to the console operator as well as an optional message is specified in the FLMGCON area.

7.14.7 Interface with TP Component

Program modifications to both TP Monitor and TP Supervisor routines were required to provide the necessary linkage needed to interface with an installation validation subroutine. TPRECORD, the routine which processes the input line and generates the accounting data, provides the interface to the installation validation subroutine at terminal LOGON time. A new subroutine function in the TPSUPEX routine was provided to interface with the installation validation subroutine at file open time. In QUIP, no further checking of file access is performed; in SODA the R,W,B indicator is checked at file update time.

An overview of the LOGON validation processing by TPRECORD is given in figure 1.

An overview of the file access validation processing done in TPSUPEX is given in figure 2.

An example of a LOGON and file access validation routine is given in figure 3.

7.14.8 Console Operator's Control of Enabling and Disabling Terminals

The console operator can place terminals allocated to NIPS/TP in an active or inactive status by replaying to the NIPS TP outstanding request. The outstanding request is:

nn, TP STANDING REQUEST. REPLIES ARE 'ENA' 'DISA' 'MSG' 'PRTY' 'TPS' OR 'TPM'.

If ENA is specified it must be followed by the designation of ALL or the name of the terminal to be enabled. More than one terminal name may be specified by enclosing the name list in parenthesis. Upon successful completion of the request, the status message TERMINAL(S) ENABLED is sent to the console operator and the advisory message TERMINAL ENABLED - ENTER LOGON is issued to the terminal. If DISA is specified it must be followed by the designation of ALL or the name of the terminal to be disabled. More than one terminal name may be specified by enclosing the name list in parenthesis. Upon successful completion of the request, the status message TERMINAL(S) DISABLED is sent to the console operator and the advisory message TERMINAL DISABLED UNTIL FURTHER NOTICE is issued to the terminal. When a terminal is in an inactive status, any attempt to enter data by the terminal user will result in the appearance of the following message: NO INPUT ALLOWED - TERMINAL IS IN DISABLED STATUS. When NIPS/TP is initialized all terminals defined in the JCL run deck are considered to be active.

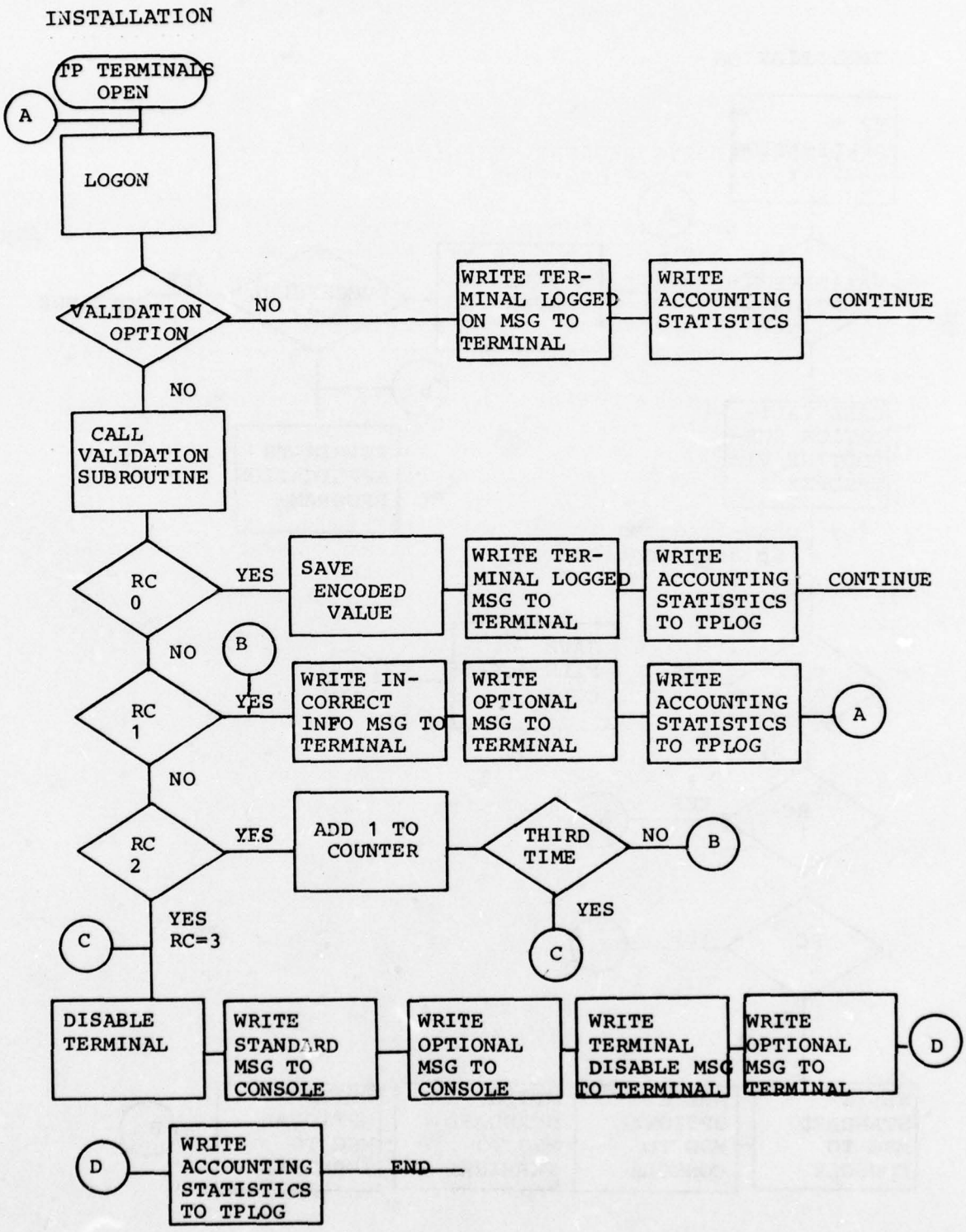


Figure 1. LOGON Validation with Installation Validation Subroutine

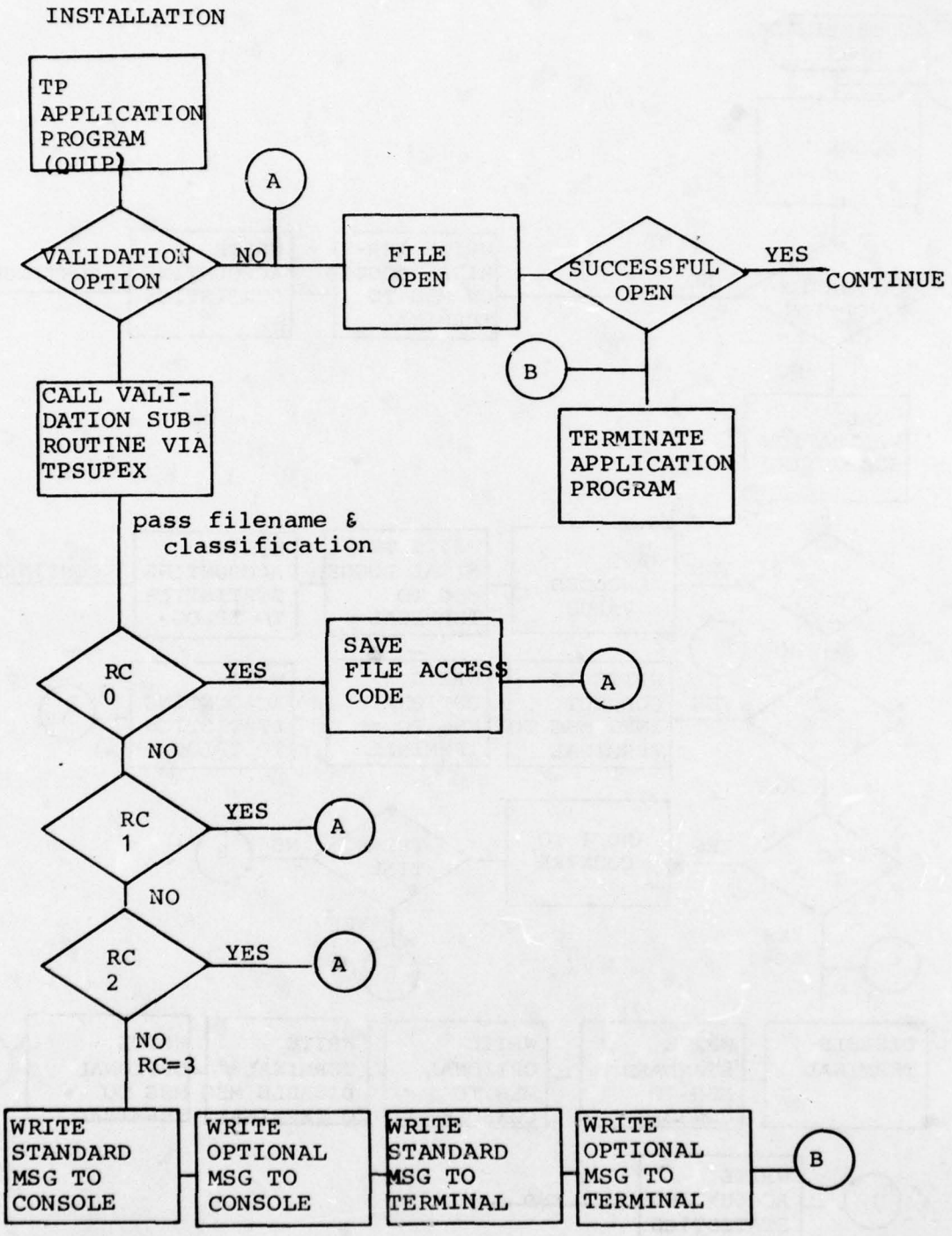


Figure 2. File Open Validation With Installation Validation Subroutine

INSTALLATION

```
                TITLE 'VALTEST - SAMPLE VALIDATION SUBROUTINE'
*
*    THIS ROUTINE ILLUSTRATES HOW AN INSTALLATION ROUTINE COULD
*    CONTROL RIGHTFUL ACCESS TO TP TERMINAL DEVICES AND DATA
*    FILES
*         UPON ENTRY REG 1 = ADDR OF PARAMETER LIST
*         PARAMETER LIST CONTAINS
*              ADDR OF SUBSUP
*              HO BYTE INDICATOR FOR LOGON OR FILE
*              ADDR OF USER INFORMATION
*              ADDR OF RETURN CODE CELL (1 CHARACTER)
*
VALTEST  CSECT
         FPSENTER  VALTEST
         BALR      2,0
         USING     *,2
         LR        6,1                  PICK-UP PARAMETER LIST
         USING     PARMLIST,6
         CLI       P2CODE,C'L'          LOGON VALIDATION
         BE        LOGON                YES-BRANCH
         CLI       P2CODE,C'F'          FILE ACCESS VALIDATION
         BE        FILE                 YES-BRANCH
         LA        15,16                SET RETURN FOR INVALID
         B         RETURN
LOGON    EQU       *
         L         7,P2CODE             PICK-UP USER LOGON INFO
         USING     TPLGDEF,7
         LA        8,LGDATA             START ADDR OF LOGON
         LA        9,70                 LENGTH OF SCAN
SCAN     CLI       0(8),C' '            TEST FOR BLANK
         BNE       CHARFIND             NO-BRANCH
         LA        8,1(8)               BUMP TO NEXT CHARACTER
         BCT       9,SCAN               DECREMENT LOOP COUNTER
         B         ERR1                 NO INFO-TREAT AS ERROR LEVEL 1
CHARFIND CLI       0(8),C'0'            TEST FOR ALPHA
         BL        ALPHA                YES-BRANCH
         B         ERR2                 NUMERIC CHARACTERS-ERROR LEVEL 2
ALPHA    LA        4,NAMETAB            LOAD NAME TABLE
         LA        5,4                  NUMBER OF ENTRIES
LOGLOOP  CLC       0(6,8),0(4)
```

Figure 3. LOGON and File Access Validation Routine (Part 1 of 3)

```
         BE        MATCH
         LA        4,10(4)                BUMP TO NEXT ENTRY
         BCT       5,LOGLOOP
         B         ERR3                   NOT IN NAME TABLE-ERROR LEVEL 3
MATCH    MVC       LGCODE,6(4)            MOVE LOGON ENCODED VALUE
         SR        15,15                  SET RETURN CODE TO ZERO
         B         RETURN
ERR1     EQU       *
         MVC       LGMGTERM(28),=C'OPTIONAL MSG - ERROR LEVEL 1'
         LA        15,1                   SET RETURN CODE TO ONE
         B         RETURN
ERR2     EQU       *
         MVC       LGMGTERM(28),=C'OPTIONAL MSG - ERROR LEVEL 2'
         LA        15,2                   SET RETURN CODE TO TWO
         B         RETURN
ERR3     EQU       *
         MVC       LGMGTERM(28),=C'OPTIONAL MSG - ERROR LEVEL 3'
         MVC       LGMGCON(38),=C'OPTIONAL LOGON MSG TO CONSOLE OPERATOR'
         LA        15,3                   SET RETURN CODE TO THREE
         B         RETURN
FILE     EQU       *
         L         7,P2CODE               PICK-UP FILE OPEN INFO
         USING     TPFLDEF,7
         CLC       FLLGCODE,=C'ALOK'      CHECK FOR SYS MRG ACCESS CODE
         BE        FMATCH
         MVC       TEMPWK+(4),FLLGCODE
         MVC       TEMPWK+4(7),FLNAME
         CLC       FLNAME(7),=C'SCIDATA'
         BNE       OLDWAY
         CLC       FLMEN(6),=C'SITREP'
         BNE       MEMOK
         MVC       FLMGTERM(31),=C'ACCESS NOT ALLOWED THIS MEMBER'
         LA        15,3
         B         RETURN
MEMOK    MVC       FLACES,=C'RM  '
         SR        15,15
         B         RETURN
OLDWAY   EQU       *
```

Figure 3. LOGON and File Access Validation Routine (Part 2 of 3)

```
         LA        4,FILETAB           LOAD FILE TABLE
         LA        5,9                 LOAD FILE TABLE
FILELOOP CLC       TEMPWK,0(4)
         BE        F1MATCH
         LA        4,16(4)             BUMP TO NEXT ENTRY
         BCT       5,FILELOOP
         MVC       FLMGTERM(33),=C'OPTIONAL MSG FROM FILE VALIDATION'
         MVC       FLMGCON(35),=C'OPTIONAL FILE ACCESS MSG TO CONSOLE'
         LA        15,3                SET RETURN CODE TO THREE
         B         RETURN
FMATCH   MVC       FLACES,=C'BALL'
         B         RET1
F1MATCH  MVC       FLACES,12(4)        MOVE FILE ACCESS CODE
RET1     SR        15,15
RETURN   FFSEXIT   RC=(15)
TEMPWK   DC        CL11' '
NAMETAB  DC        C'BROWN LGAA'
         DC        C'JONES LGBB'
         DC        C'SMITH LGBC'
         DC        C'SYSMGRALOK'
FILETAB  DC        C'LGAATESTER  RFLA'
         DC        C'LGAATRAINER RFLA'
         DC        C'LGAAMENUDATARA '
         DC        C'LGAASCIDATA RM '
         DC        C'LGBBTESTER  WFLB'
         DC        C'LGBBTRAINER BFLB'
         DC        C'LGBBXTEST   RFLB'
         DC        C'LGBBMESTER  RFLB'
         DC        C'LGCCTRAINER RFLB'
PARMLIST DSECT
P1ADDR   DS        A
P2CODE   DS        CL1
P2ADDR   DS        AL3
P3ADDR   DS        A
         QTPFLDEF
         QTPLGDEF
         END
```

Figure 3. LOGON and File Access Validation Routine (Part 3 of 3)

7.15 TPQUIPVS Load Structure for QUIP

QUIP is unique program in the NIPS TP component in that by choosing the QUIP load structure for the installation it is possible to tailor the Monitor and Supervisor generation to the resources available and the user requirements for query response time.

The distributed TP Monitor and Supervisor are generated to use the TPQUIP load structure for QUIP. This is the one used most commonly since it was the only one available for quite some time. In the TPQUIP load structure, QUIP consists in a number of distinct load modules, of various sizes, which are linked to or loaded and called as required during translation and execution of a query. This structure allows QUIP to execute with a minimum of core requirements since only routines necessary for the particular query are brought into core, and then only for the duration of use. The disadvantage of this structure is the overhead needed to load and delete the various modules as they are used.

The TPQUIPVS load structure has been designed to eliminate that overhead for certain types of queries. All the QUIP load modules which process the QUIP operators LIMIT, retrieval IF, LIST, and LOAD have been combined into a single load module to make them core resident for as long as at least one QUIP user is signed on. Significant overhead reductions are also possible for queries which process other operators since all the load modules which are required for the execution of every query have also been included in the TPQUIPVS structure.

By designating the TPQUIPVS load structure for QUIP it is possible to achieve automatic reduction in QUIP setup time (i.e., the elapsed time in QUIP from the start of the query to the start of file search) as compared to the time obtained with the TPQUIP structure. Therefore, if queries can be designed so that record qualification and data output begin with the start of file search (e.g., the LIMIT operator restricts retrieval to those records which contain data to be displayed), improved response time will be obtained for the time the first page of data is received on the CRT. Additional improvement is possible when the

INSTALLATION

INCORFFT option is specified for QUIP in TP Supervisor generation (see section 7.5, TP Supervisor Generation).

Appendix A

ALLOCATION OF TP MESSAGE QUEUES

This appendix provides sample JCL to allocate the message queues required by TP. The JCL reflects the defaults names for the Output Message Queues.

A.1 TPQ Job

```
//TPQ     JOB
//       EXEC   PGM=IEFBR14
//DD1     DD    DSNAME=TPIMQ,DISP=(NEW,CATLG),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(80,501)
//DD2     DD    DSNAME=T.DD52250A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD3     DD    DSNAME=T.D512250A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFLIB,SPACE=(TRK,(20))
//DD4     DD    DSNAME=T.DD62260A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD5     DD    DSNAME=T.DD62260B,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD6     DD    DSNAME=T.DD62260C,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD7     DD    DSNAME=T.DD62260D,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD8     DD    DSNAME=T.DD62260E,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD9     DD    DSNAME=T.DD62260F,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD10    DD    DSNAME=T.DD62260G,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DD11    DD    DSNAME=T.DD62260H,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE2    DD    DSNAME=E.DD52250A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE3    DD    DSNAME=E.D512250A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFLIB,SPACE=(TRK,(20))
//DDE4    DD    DSNAME=E.DD62260A,DISP=(NEW,KEEP),UNIT=2314,
//             VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
```

```
//DDE5    DD    DSNAME=E.DD62260B,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE6    DD    DSNAME=E.DD62260C,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE7    DD    DSNAME=E.DD62260D,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE8    DD    DSNAME=E.DD62260E,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE9    DD    DSNAME=E.DD62260F,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE10   DD    DSNAME=E.DD62260G,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
//DDE11   DD    DSNAME=E.DD62260H,DISP=(NEW,KEEP),UNIT=2314,
//              VOLUME=SER=FFSLIB,SPACE=(TRK,(20))
/*
```

Appendix B

TYPICAL INSTALLATION PROCEDURE

This appendix provides more specific information to the systems programmer who intends to install NIPS 360 FFS at a new installation. It suggests a sequence of steps to install the system and describes the jobs to be performed at each step. JCL examples for the jobs discussed are found in appendix C. It is intended that this typical procedure will provide additional insight by illustrating the type of JCL required to install NIPS 360 FFS.

B.1 Clip Disk Pack to FFSLIB

Run the PACKLABL job in the Installation Package to provide a disk pack labeled FFSLIB. If the IEHDASDR Utility program is not available, run a stand-alone CLIP program. To allow a minimum amount of changes to the JCL for the jobs in the Installation Package, all references to disk space were made with a UNIT=2314 and VOLUME=SER=FFSLIB.

Ideally, if one scratch pack can be reserved for this purpose, installation and testing of the FFS system can be performed in the most optimum manner since the sample JCL assumes a pack labeled FFSLIB. After installation, the system data sets can always be moved to other packs and recataloged, returning this pack to the installation. The scratch pack is assumed to be a 2314.

B.2 Catalog System Data Sets

To minimize the amount of JCL required in subsequent jobs, the following data sets are cataloged to FFSLIB using the CATLG job:

FFS.JOBLIB
FFS.JOBMACRO
FFS.PROCLIB
PTF.JOBLIB
PTF.JOBMACRO
TESTERL
DUMMY.FILE
DUMMY.FILEL
DUMMY.FILES
DUMMY.FILEX

The unit type for the pack labeled FFSLIB is assumed to be a 2314. The data sets DUMMY.FILES and DUMMY.FILEX are cataloged to the pack labeled FFSLIB. These data sets are necessary for Procedures JCL resolution but are never actually used.

B.3 Restore System Data Sets

The following data sets must be loaded to direct access:

FFS.JOBLIB
FFS.JOBMACRO
FFS.PROCLIB
PTF.JOBLIB
PTF.JOBMACRO
DUMMY.FILEL
TESTERL

The data set FFS.PROCLIB is restored to update SYS1.PROCLIB and can subsequently be uncataloged and scratched. TESTERL is only needed to run the sample jobs.

If the system data sets were received on a dump/restore tape, the restore operation becomes relatively simple. The RESTORE job will load all data sets onto FFSLIB for installations having the IEHDASDR Utility program.

If the system data sets were received as sequential and unloaded partitioned data sets, space is allocated for each data set on FFSLIB and the data sets reloaded to the pack. The RELOAD job should accomplish this task. This job is set up for a 2314 reload; if 2311's are being used, triple the primary allocation for each data set (except DUMMY.FILEL).

B.4 Update SYS1.PROCLIB

The COPYPROC job could be used to place the FFS procedures in the SYS1.PROCLIB. If a source copy of the procedures is desired, the IEBPTPCH Utility program will punch a copy of these procedures. An IEBUPDTE Utility program could then place the procedures into the SYS1.PROCLIB with the punched cards as input.

The distributed procedures must be modified at some installations. For example, if the operating system at an installation has been generated for UNIT values other than NIPW, TAPE9 and TAPE7, the procedures should be modified to conform to installation conventions. The procedures can be punched out from FFS.PROCLIB, modified, and then placed on SYS1.PROCLIB. They can also be changed on FFS.PROCLIB and then moved to SYS1.PROCLIB with the IEBUPDTE utility program.

B.5 Run the Sample Job

The sample job is distributed as an unblocked card image data set. This partitioned data set is prepared with the sample jobs in the sequence suggested in Section 3.9, Sample Jobs Library. The data file created by the FFTJOB job will be placed on the same disk pack that the Sample Subroutine Library (TESTERL) resides.

The TESTERL library also contains subroutines developed by other installations. These subroutines are distributed with the NIPS system as they may be of use to other NIPS users. If these subroutines are being used, the TESTERL library should not be scratched.

The following JCL may be used to punch out the sample job JCL statements:

```
//SAMPLJOB JOB
//PUNCH EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=OLD,DSN=NIPS.SAMPLE.JOB(BLDJCL)
//SYSUT2 DD SYSOUT=B
/*
```

Each job of the sample jobs tape has a job card with the following format:

```
//JOBNAME JOB,MSGLEVEL=1,PRTY=NN
```

If the installation requires specific accounting information, the sample jobs can be modified and run individually in the sequence suggested by the priorities assigned to each job.

The JCL required to create the NIPS system is contained in appendix C. The JCL is distributed on the system tape as a sequential data set consisting of card images. The OS utility IEBGENER may be used to punch this data set, so that the JCL can be modified to reflect installation and/or system requirements. The following job may be used to punch the installation JCL:

```
//NIPSJCL JOB
//PUNCH EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=OLD,DSN=NIPSJCL
//SYSUT2 DD SYSOUT=B
//SYSIN DD DUMMY
/*
```

Appendix C

TYPICAL INSTALLATION JCL

In the following JCL examples, the JCL statements begin in card column 1, the continuation character X is assumed to be in card column 72, and all continuation cards begin in card column 16.

C.1 PACKLABL Job

This job will provide a volume serial number of FFSLIB to a 2314 disk pack currently labeled XXXXXX. It is recommended that this pack be initialized (DASDI) before using as a NIPS 360 FFS system pack.

If the IEHDASDR Utility program is not available with the current release of your Operating System, a stand-alone CLIP program could be used.

```
//PACKLABL JOB
//GO       EXEC     PGM=IEHDASDR
//SYSPRINT DD       SYSOUT=A
//DISK     DD       DISP=OLD,UNIT=(2314,,DEFER),
//                 VOLUME=(PRIVATE,,SER=(XXXXXX))
//SYSIN    DD       *,DCB=BLKSIZE=80
         LABEL    TODD=DISK,NEWVOLID=FFSLIB

/*
```

C.2 CATLG Job

This job will catalog NIPS 360 FFS data sets to a 2314 disk pack labeled FFSLIB.

```
//CATLG    JOB
//GO       EXEC        PGM=IEHPROGM
//SYSPRINT DD          SYSOUT=A
//SYSTEM   DD          DSNAME=SYS1.SVCLIB,DISP=OLD
//SYSIN    DD          *,DCB=BLKSIZE=80
     CATLG  DSNAME=FFS.JOBLIB,VOL=2314=FFSLIB
     CATLG  DSNAME=FFS.JOBMACRO,VOL=2314=FFSLIB
     CATLG  DSNAME=FFS.PROCLIB,VOL=2314=FFSLIB
     CATLG  DSNAME=PTF.JOBLIB=VOL=2314=FFSLIB
     CATLG  DSNAME=PTF.JOBMACRO,VOL=2314=FFSLIB
     CATLG  DSNAME=TESTFPL,VOL=2314=FFSLIB
     CATLG  DSNAME=DUMMY.FILE,VOL=2314=FFSLIB
     CATLG  DSNAME=DUMMY.FILEL,VOL=2314=FFSLIB
     CATLG  DSNAME=DUMMY.FILES,VOL=2314=FFSLIB
     CATLG  DSNAME=DUMMY.FILEX,VOL=2314=FFSLIB
     CATLG  DSNAME=DUMMY.ISAMFILE,VOL=2314=CANCEL
     CATLG  DSNAME=DUMMY.SAMFILE,VOL=2314=CANCEL
/*
```

Note: DUMMY.ISAMFILE and DUMMY.SAMFILE are not required to be cataloged if there is no TP support.

C.3 RESTORE Job

This job will restore all NIPS 360 FFS data sets to a 2314 disk pack labeled FFSLIB. It should be used with the IEHDASDR Utility program.

```
//RESTORE      JOB
//            EXEC  PGM=IEHDASDR
//SYSPRINT     DD    SYSOUT=A
//TAPE         DD    DSNAME=FFS360,DISP=OLD,
//                   LABEL=(,SL),DCB=DEN=2,
//                  UNIT=(2400,,DEFER),VOL=(PRIVATE,,SER=(FFS360))
//DISK         DD    UNIT=2314,VOL=SER=FFSLIB,DISP=OLD
//SYSIN        DD    *
     RESTORE      TODD=DISK,FROMDD=TAPE,PURGE=YES
/*
```

C.4 RELOAD Job

This job will restore the NIPS 360 FFS data sets that are distributed as unloaded partitioned and sequential data sets. The Skeleton File Library DUMMY.FILEL is created but no data is moved into this data set.

```
//RELOAD   JOB
//ALLOCATE EXEC    PGM=IEFBR14
//DD1      DD      DSNAME=FFS.JOBLIB,DISP=(,KEEP),UNIT=2314,     X
//                 SPACE=(CYL,(30,5,50)),VOLUME=SER=FFSLIB,      X
//                 DCB=(BLKSIZE=7294,RECFM=U)
//DD2      DD      DSNAME=FFS.JOBMACRO,DISP=(,KEEP),UNIT=2314,   X
//                 SPACE=(CYL,(10,5,50)),VOLUME=SER=FFSLIB,      X
//                 DCB=(BLKSIZE=3360,LRECL=80,RECFM=FB)
//DD3      DD      DSNAME=PTF.JOBLIB,DISP=(,KEEP),UNIT=2314,     X
//                 SPACE=(CYL,(10,5,50)),VOL=SER=FFSLIB,         X
//                 DCB=(BLKSIZE=7294,RECFM=U)
//DD3A     DD      DSNAME=PTF.JOBMACRO,DISP=(,KEEP),UNIT=2314,   X
//                 SPACE=(CYL,(2,2,10)),VOL=SER=FFSLIB,          X
//                 DCB=(BLKSIZE=3360),LRECL=80,RECFM=FB
//DD4      DD      DSNAME=FFS.PROCLIB,DISP=(,KEEP),UNIT=2314,    X
//                 SPACE=(CYL,(5,1,50)),VOLUME=SER=FFSLIB,       X
//                 DCB=(BLKSIZE=80,LRECL=80,RECFM=FB)
//DD5      DD      DSNAME=TESTERL,DISP=(,KEEP),UNIT=2314,        X
//                 SPACE=(CYL,(2,2,10)),VOLUME=SER=FFSLIB,       X
//                 DCB=(BLKSIZE=7294,RECFM=U)
//DD6      DD      DSNAME=DUMMY.FILEL,DISP=(,KEEP),UNIT=2314,    X
//                 SPACE=(CYL,(1,1,1)),VOLUME=SER=FFSLIB,        X
//                 DCB=(BLKSIZE=7294,RECFM=U)
//LOAD     EXEC    PGM=IEHMOVE
//SYSPRINT DD      SYSOUT=A
//SYSUT1   DD      DISP=OLD,UNIT=2314,VOLUME=SER=FFSLIB
//DD1      DD      DISP=OLD,UNIT=2314,VOLUME=SER=FFSLIB
//TAPIN    DD      DSNAME=FFS360,UNIT=2400,DISP=(,KEEP),         X
//                 LABEL=(1,SL),VOLUME=SER=FFS360,               X
//                 DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DEN=2)
//SYSIN    DD      *,DCB=BLKSIZE=80
    COPY PDS=FFS.JOBLIB,TO=2314=FFSLIB,FROM=2400=(FFS360,1)
    COPY PDS=FFS.JOBMACRO,TO=2314=FFSLIB,FROM=2400=(FFS360,2)
```

```
    COPY PDS=FFS.PROCLIB,TO=2314=FFSLIB,FROM=2400=(FFS360,3)
    COPY PDS=TESTERL,TO=2314=FFSLIB,FROM=2400=(FFS360,4)
    COPY PDS=NIPS.SAMPLE.JOB,TO=2314=FFSLIB,FROM=2400=(FFS360,5)
    COPY PDS=PTF.JOBLIB,TO=2314=FFSLIB,FROM=2400=(FFS360,6)
    COPY PDS=PTF.JOBMACRO,TO=2314=FFSLIB,FROM=2400=(FFS360,7)
/*
```

C.5 COPYPROC Job

This job will update SYS1.PROCLIB from the NIPS 360 FFS data set named FFS.PROCLIB. The data set resides on a 2314 disk pack labeled FFSLIB and is assumed to be cataloged.

```
//COPYPROC JOB
//GO            EXEC     PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD       SYSOUT=A
//SYSUT1   DD       DSNAME=FFS.PROCLIB,DISP=OLD,UNIT=2314,  X
//                  VOLUME=SER=FFSLIB
//SYSUT2   DD       DSNAME=SYS1.PROCLIB,DISP=OLD
//SYSIN    DD       *,DCB=BLKSIZE=80
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XFFSPTFL
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XFM
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XFMEX
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XFR
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XFS
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XISTOS
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XOP
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XOPEX
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XOPSD
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XOPSDEX
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XRASP
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XRASPEX
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSTOIS
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSUBLDR
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XTABGEN
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XQUIP
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XQUIPSD
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XQRTQDF
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSAVEANS
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XRESTANS
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSAVELIB
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XRESTLIB
./         REPRO    LEVEL=00,SOURCE=0,LIST=ALL,NAME=XDMPLIB
```

```
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSUBCHK
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XCLASS
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XUTFSCAN
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XSP
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XTRTAPE
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XTRDISK
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XTP
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=TPRDR080
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XKA
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XKM
./          REPRO     LEVEL=00,SOURCE=0,LIST=ALL,NAME=XUTODE
./          ENDUP
/*
```

Note: XTP and TPRDR080 need only be included if TP is required.

Appendix D

DATA SET SPECIFICATIONS

DATA SET	*SPACE	BLKSIZE	LRECL	RECFM
FFS.JOBLIB	(CYL,(30,5,50))	7294		U
FFS.JOBMACRO	(CYC,(10,5,50))	3360	80	FB
FFS.PROCLIB	(CYL,(5,1,50))	80	80	FB
PTF.JOBLIB	(CYL,(10,5,50))	7294		U
PTF.JOBMACRO	(CYL,(2,2,10))	3360	80	FB
TESTERL	(CYCL,(2,2,10))	7294		U
DUMMY.FILEL	(CYL,(1,1,1))	7294		U

*2314 cylinders

Note: The FFS.JOBLIB, TESTERL, and DUMMY.FILEL libraries should have a blocksize of 7294 if they are to reside on a 2314 disk pack and a blocksize of 3625 if they are to reside on a 2311 disk pack.

Appendix E

DCR Number:_____

NIPS 360 PPS
DISCREPANCY/CHANGE REPORT

Name:_____Date:_____Phone:_____

Address:_____

NIPS Component(s):_____NIPS Release:_____ Accompanying Materials

OS Release:_____(circle one: PCP-MFT-MVT)_____ __Core Dump __JCL Stream

Date problem encountered:_____ __Console List __Other

Description of problem:

Actions already taken:

Received by:_____Date:_____

Time:_____Turned over to:_____

INSTALLATION

Appendix F

DCR NUMBER: _____

IAR NUMBER: _____

MAINTENANCE PROGRAMMER'S REPORT

Completed by:_____Date: _____

Type of Problem:

____Component Problem	_____Machine Malfunction
____User Problem	_____Other
____OS Problem	_____Improvement Suggestion

DISTRIBUTION

CCTC CODES	COPIES
C124 (Reference and Record)	3
C124 (Record Copy) Stock	6
C240	20
C315	1
C341 (Maintenance Contractor)	10
C341 (Stock)	70

EXTERNAL

Director of Administrative Services, Office of the Joint Chiefs of Staff
Attn: Chief, Personnel Division, Room 1A724, The Pentagon Washington, D.C. 20301 ---------- 1

Director for Personnel, J-1, Office of the Joint Chiefs of Staff, Attn: Chief, Data Service Office, Room 1B738C, The Pentagon, Washington, D.C. 20301 ---------- 1

Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: P & AD, Room 2B870, The Pentagon, Washington, D.C. 20301 ---------- 1

Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: Deputy Director for Operations (Reconnaisance and Electronic Warfare) Room 2D921, The Pentagon, Washington, D.C. 20301 ---------- 1

Director for Logistics, J-4, Office of the Joint Chiefs of Staff, Room 2E828, The Pentagon, Washington, D.C. 20301 ---------- 1

Chief, Studies Analysis and Gaming Agency, Attn: Chief, Force Analysis Branch, Room 1D928A, The Pentagon, Washington, D.C. 20301 ---------- 1

Automatic Data Processing, Liaison Office National Military Command Center, Room 2D901A, The Pentagon, Washington, D.C. 20301 ---------- 1

EXTERNAL	COPIES
Automatic Data Processing Division Supreme Headquarters Allied Powers, Europe Attn: SA & P Branch, APO New York 09055------	1
Director, Defense Communications Agency, Office Of MEECN System Engineering, Attn: Code 960T, Washington, D.C. 20301----------	1
Director, Defense Communications Engineering Center, Hybrid Simulation Facility, 1860 Wiehl Avenue, Reston, VA 22070----------	1
Director, Defense Intelligence Agency Attn: DS - 5C2 Washington, D.C. 20301----------	5
Commander-in-Chief, Pacific, Attn: J6331, FPO San Francisco, 96610----------	1
Commander-in-Chief, US Army Europe and Seventh Army ATTN: OPS APO New York 09403---	1
Commanding General, US Army Forces Command, Attn: Data Support Division, Building 206, Fort McPherson, GA 30303----------	1
Commander, Fleet Intelligence Center, Europe, Box 18, Naval Air Station, Jacksonville, Florida 32212----------	1
Commanding Officer, Naval Air Engineering Center, Ground Support Equipment Department, SE 314, Building 76-1, Philadelphia, PA 19112	1
Commanding Officer, Naval Security Group Command, 3801 Nebraska Avenue, N.W. Attn: GP22, Washington, D.C. 20390----------	1
Commanding Officer, Navy Ships Parts Control Center, Attn: Code 712, Mechanicsburg, PA 17055	1
Headquarters, US Marine Corps, Attn: System Design and Programming Section (MC-JSMD-7) Washington, D.C. 20380----------	1

EXTERNAL	COPIES
Commanding Officer, US Army Forces Command Intelligence Center, Attn: AFIC-PD, Fort Bragg, NC 28307----------	1
Commander, US Army Foreign Science and Technology Center, Attn: AMXSJ-CS, 220 Seventh Street NE, Charlottsville, VA 22212--	1
Commanding Officer, US Army Security Agency, Command Data Systems Activity (CDSA) Arlington Hall Station, Arlington, VA 22212----------	1
Commanding Officer, US Army Security Agency Field Station - Augsburg, Attn: IAEADP, APO New York 09458----------	1
Commander, Fleet Intelligence Center, Atlantic, Attn: DPS, Norfolk, VA 23511----------	1
Commander, Fleet Intelligence Center, Pacific, Box 500, Pearl Harbor, HI 96860----------	1
Air Force Operations Center, Attn: Systems Division (XOOCSC) Washington, D.C. 20301-----	1
Commander, Armed Forces Air Intelligence Training Center, TTMNIM (360 FFS), Lowry AFB, Co 80230----------	1
Commander, Air Force Data Services Center, Attn: Director of System Support, Washington, D.C. 20330----------	1
Commander-in-Chief, US Air Forces in Europe, Attn: ACDI APO New York 09332----------	1
Commander, USAF Tactical Air Command, Langley AFB, VA 23665----------	1
Commander, Space and Missile Test Center, Attn: (ROCA) Building 7000, Vandenberg, AFB, CA 93437----------	1

EXTERNAL	COPIES
Naval Air Systems Command, Naval Air Station, Code 13999, Jacksonville, Florida 32212------	1
Commanding General, US Army Computer Systems Command, Attn: Support Operations Directorate, Fort Belvoir, VA----------	1
Defense Documentation Center, Cameron Station, Alexandria, VA 22314----------	12
TOTAL	159

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR54-78	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NMCS Information Processing System 360 Formatted File System (NIPS 360 FFS) Installation of NIPS 360 FFS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA 100-77-C-0065
9. PERFORMING ORGANIZATION NAME AND ADDRESS International Business Machines, Corp. Rosslyn, Virginia		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Military Command System Support Center The Pentagon, Washington, D.C. 20301		12. REPORT DATE 1 September 1978 13. NUMBER OF PAGES 173
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This Technical Report describes the installation and maintenance of the NIPS 360 Formatted File System. It describes the hardware and software requirements of the S/360 Operating System (OS). The functions to be performed to install NIPS 360 FFS are presented and a detailed typical 2314 installation is included as a guideline. The procedures for reporting system deficiencies and for periodically updating the system to incorporate system improvements are included. This document supersedes TR 54-77.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)